THE BELL SYSTEM TECHNICAL JOURNAL

JULY-AUGUST 1983
VOL. 62, NO. 6, PART 1

On the Start-up Problem in Digital Echo Cancelers 1353
J. Salz

Sample Reduction and Subsequent Adaptive Interpolation of Speech Signals 1365
R. Steele and F. Benjamin

Practical Design Considerations for Coupled-Single-Amplifier-Biquad Active Bandpass Filters 1399
J. Tow

Modal Structure of an MCVD Optical Waveguide Fiber 1415
A. Carnevale and U. C. Paek

Asymptotic Analysis of a Queueing Model With Bursty Traffic 1433
D. Y. Burman and D. R. Smith

MODERNIZATION OF THE SUBURBAN ESS

Overview: Evolution of the Suburban ESS 1455
T. E. Grassman and J. E. Yates

Adding Data Links to an Existing ESS 1467
C. E. Ishman, R. B. Sanderson, L. M. Taff, D. P. Truax, and C. T. Tulloss

Hosting the No. 10A Remote Switching System 1497
D. W. Brown, J. J. Driscoll, F. M. Lax, M. W. Saad, and J. G. Whitemyer

Billing and Measurements Modernization 1537
J. P. Lodwig and D. A. Ward

ACRONYMS AND ABBREVIATIONS 1551

PAPERS BY BELL LABORATORIES AUTHORS 1553

CONTENTS, SEPTEMBER ISSUE 1557

THE BELL SYSTEM TECHNICAL JOURNAL

THE BELL SYSTEM TECHNICAL JOURNAL (ISSN0005-8580) is published by the American Telephone and Telegraph Company, 195 Broadway, N. Y., N. Y. 10007; C. L. Brown, Chairman and Chief Executive Officer; W. M. Ellinghaus, President; V. A. Dwyer, Vice President and Treasurer; T. O. Davis, Secretary.

The Journal is published in three parts. Part 1, general subjects, is published ten times each year. Part 2, Computing Science and Systems, and Part 3, single-subject issues, are published with Part 1 as the papers become available.

The subscription price includes all three parts. Subscriptions: United States—1 year $35; 2 years $63; 3 years $84; foreign—1 year $45; 2 years $73; 3 years $94. Subscriptions to Part 2 only are $10 ($12 foreign). Single copies of the Journal are available at $5 ($6 foreign). Payment for foreign subscriptions or single copies must be made in United States funds, or by check drawn on a United States bank and made payable to The Bell System Technical Journal and sent to Bell Laboratories, Circulation Dept., Room 1E-335, 101 J. F. Kennedy Parkway, Short Hills, N. J. 07078.

Comments on the technical content of any article or brief are welcome. These and other editorial inquiries should be addressed to the Editor, The Bell System Technical Journal, Bell Laboratories, Room 1J-319, 101 J. F. Kennedy Parkway, Short Hills, N. J. 07078. Comments and inquiries, whether or not published, shall not be regarded as confidential or otherwise restricted in use and will become the property of the American Telephone and Telegraph Company. Comments selected for publication may be edited for brevity, subject to author approval.

Printed in U.S.A. Second-class postage paid at Short Hills, N. J. 07078 and additional mailing offices. Postmaster: Send address changes to The Bell System Technical Journal, Room 1E-335, 101 J. F. Kennedy Parkway, Short Hills, N. J. 07078.

THE BELL SYSTEM
TECHNICAL JOURNAL

DEVOTED TO THE SCIENTIFIC AND ENGINEERING
ASPECTS OF ELECTRICAL COMMUNICATION

Volume 62 July-August 1983 Number 6, Part 1

On the Start-Up Problem in Digital Echo Cancelers

By J. SALZ*

(Manuscript received October 5, 1982)

Digital echo cancellation techniques make it possible to realize efficient full-duplex data transmission over a single loop. The purpose of this paper is to elucidate the solution to the start-up problem in these devices and to present a new, fast, and simple tap-adjustment procedure. The theory indicates that a modified stochastic gradient tap-adjustment algorithm, using pseudorandom input data sequences for the initial training period, converges in N steps, where N is the total number of canceler taps, and that this is the fastest possible convergence time.

I. INTRODUCTION

Two-way voice communication over a single loop is made possible by the use of a hybrid bridge. However, the suppression of echoes by fixed hybrids is insufficient to support full-duplex data transmission, and therefore makes adaptive data echo cancelers necessary.

* Bell Laboratories.

In two-way data communications, transmitted data "echoes" back to the near-end receiver after being reflected and dispersed through an unknown return path. So, if one assumes that the echo path is linear, the estimation of its overall impulse response is sufficient to allow the synthesis of one's own echo signal. This synthesized version is subtracted from the received signal, which then makes it possible for the receiver to extract the data intended for it.

The impulse response of the echo channel (local transmitter output to local receiver input) can be measured in several ways. An obvious way to do this is to transmit a single impulse and measure the echo. However, among other defects of this procedure, the average power would be very low and the resulting signal-to-noise ratio (s/n) would be inadequate. If a pseudorandom sequence (+1 or −1) is transmitted instead, the average power would be much greater and would be essentially constant on the line, more nearly representing a true data signal. The latter is the preferable approach.

Digital data echo cancelers operate in two modes. In the acquisition mode, or start-up, the impulse response of the echo path is measured. This is best accomplished, as we shall see, with the use of fixed data sequences. Since, during this period, no information is conveyed to the far-end, the time allotted for this purpose should be as short as possible. Although it is conceptually possible to start an echo canceler either blind or with random data, the convergence of the taps, or the reliable measurements of the impulse response, are known to require a long time. In the subsequent mode, or during actual data transmission, a tracking algorithm is initiated whose function is to update the measurements when slight changes occur in the impulse response. We focus on the more critical start-up algorithm.

From an operational point of view, it is desirable to implement these algorithms in a recursive fashion, or in a closed-loop manner. This means that the canceler tap coefficients, which represent the sampled impulse response, are updated in response to a measured error between the actual impulse response and the one estimated at any particular instant. Conventional gradient adjustment algorithms, even with fixed data sequences, are known to converge very slowly.

This research was motivated by the need for a theory capable of explaining the behavior of tap-adjustment algorithms. During the course of this investigation a modified stochastic gradient algorithm that is simple to implement and converges in the theoretically smallest number of steps was discovered.

II. PROBLEM FORMULATION

Figure 1 shows a full-duplex data modem employing a digital echo canceler. We are concerned with the sampled signal values

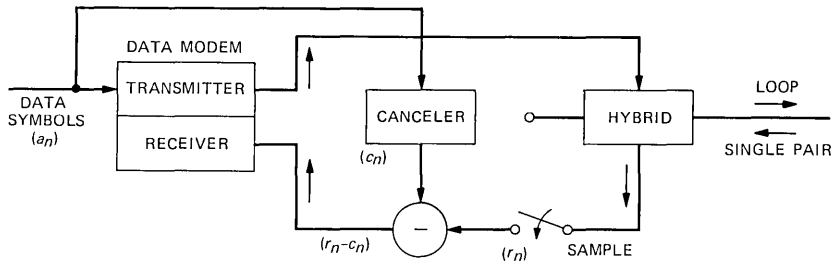


Fig. 1—Digital data canceler.

$$r_n = \sum_{k=-\infty}^{\infty} h_k a_{n-k} + \nu_n, \tag{1}$$

where the a_n's are the transmitted data symbols that echo back to the receiver, the h_n's are the overall impulse response values of the echo path, and the ν_n's are the desired received signal values plus noise. The object of the canceler is to synthesize signal values, c_n, which are estimates of $\sum_k h_k a_{n-k}$ and subtract them from the r_n's. The receiver then proceeds to process the difference signal values, $r_n - c_n$, to extract the data intended for it.

The fundamental problem is to devise procedures for estimating the h_n's from observations of the r_n's, while treating the ν_n's as undesirable noise. Practically, it must be assumed that only a finite number, N, of the h_n's can be estimated, and so we express (1) compactly as

$$r_n = H^t A_n + \nu_n, \tag{2}$$

where H and A_n are finite dimensional column vectors,*

$$H = \begin{pmatrix} h_N \\ \vdots \\ h_1 \end{pmatrix}, \quad \text{and} \quad A_n = \begin{pmatrix} a_{n-N} \\ \vdots \\ a_{n-1} \end{pmatrix},$$

and where $(\,)^t$ indicates the transpose of a matrix.

In the absence of precise statistical knowledge of the ν_n's and H, a natural procedure for choosing an estimator of H is to minimize the sum of squared errors from time 1 to time l

$$\epsilon_l = \sum_{n=1}^{l} (r_n - \hat{H}^t A_n)^2.$$

This is a standard problem and the solution is immediate. It involves the solution of a set of linear equations

$$Z_l \hat{H}_l = U_l \tag{3}$$

* We deal with a baseband model for notational convenience. By using complex numbers throughout, the treatment generalizes to passband models.

for the best estimator at time l, $\hat{H}_l$.

In (3)
$$U_l = \sum_{n=1}^{l} A_n r_n$$

and
$$Z_l = \sum_{n=1}^{l} A_n A_n^t.$$

In applications it is usually desirable to solve these equations recursively with a minimum computational effort and to assure rapid convergence to the actual H. Our attention in the next sections is directed toward these aims. However, before dealing with our main subject we wish first to examine some asymptotic behaviors of the standard solution.

2.1 Random input data

In some applications, echo cancelers must start blind, i.e., from random data. This is the case in echo cancelers used to suppress speech. However, in full-duplex data communications, a preamble word, or words, can be sent first to assure rapid convergence. To emphasize these differences we examine the behavior of the estimator with random data first, where one has no choice in the selection of the starting sequences. So, consider random data such that the a_n's assume ± 1 independently with equal probability and examine the limit as $l \to \infty$. It is found that

$$\begin{aligned} U_l &\to lE\{A_n r_n\} \\ &= lE\{A_n A_n^t\}H, \\ Z_l &\to lE\{A_n A_n^t\} = lI, \end{aligned}$$

and, consequently,

$$Z_l^{-1} U_l \to H. \tag{4}$$

In the above we made use of the fact that the v_n's and A_n's are naturally independent. We assumed that these sequences are ergodic and so replaced time averages with mathematical expectations, $E\{\cdot\}$. This then demonstrates that if one is willing to wait forever, it is conceptually possible to determine H exactly—not a terribly startling result.

While the asymptotic behavior is easy to deduce, the statistical behavior of the estimator for finite l is difficult to glean. One immediately encounters an unsolved mathematical problem that involves the conditions on the random sequences that would guarantee the existence of the inverse matrix Z_l^{-1}. Clearly, l has to be greater than N for the inverse to even have a chance to exist, and for those

sequences for which the inverse exists, we can claim that the estimator is unbiased. This can be seen by the direct computation

$$\hat{H}_l = H + Z_l^{-1}\left(\sum_{n=1}^{l} A_n \nu_n\right) \tag{5}$$

and so

$$E\hat{H}_l = H,$$

since the ν_n's are assumed to have zero mean value. We now turn our attention to the behavior of the solution with selected fixed sequences.

2.2 *Fixed sequences*

Here the A_n's can be chosen to ensure the existence of Z_l^{-1}, and so in addition to establishing that the estimator is unbiased, we can now also calculate the error matrix

$$\begin{aligned} R &= E\{(\hat{H}_l - H)(\hat{H}_l - H)^t\} \\ &= E\left[Z_l^{-1}\left(\sum_{n,m=1}^{l} A_n \nu_n \nu_m A_m^t\right) Z_l^{-1}\right] \\ &= \sigma^2 Z_l^{-1}\left(\sum_{n=1}^{l} A_n A_n^t\right) Z_l^{-1} \\ &= \sigma^2 Z_l^{-1}, \end{aligned} \tag{6}$$

where we again assume that the ν_n's are independent with variance = σ^2. From this we obtain the variance of the estimator

$$\sigma_H^2 = \text{Trace } R = \sigma^2 \text{ Trace}\left(\sum_{n=1}^{l} A_n A_n^t\right)^{-1}, \tag{7}$$

where Trace $(\cdot)$ stands for the trace of a matrix. As can be verified again from (7), $\sigma_H^2 \to 0$, $l \to \infty$, as it should.

Evidently, the value of σ_H^2 depends, in a critical way, on the detailed structure of the data sequences, and so the choice of the particular sequences becomes important. The evaluation of (7) can always be carried out once the data sequences are specified. To illustrate the kind of behavior that can be expected and to set the stage for the next section, we evaluate this expression for two sets of sequences, orthogonal and pseudorandom.

For $l = N$ the ijth element of Z_N, which is composed of orthogonal sequences, A_n, is

$$\left(\sum_{n=1}^{N} A_n A_n^t\right)_{ij} = \sum_{n=1}^{N} a_{n-i} a_{n-j} = \begin{cases} N, & i = j \\ 0, & i \neq j \end{cases} \tag{8}$$

and therefore $\sigma_H^2 = \sigma^2$. Clearly, orthogonal sequences yield the very best result possible, but it is not known how to construct these sequences for arbitrary N. Fortunately, pseudorandom periodic sequences, which are generated by linear shift registers, exhibit very desirable properties for our application. We shall discuss these sequences and their properties in greater detail in the next section. For now suffice it to state that it is well known how to generate sequences with the property

$$(Z_N)_{ij} = \sum_{n=1}^{N} a_{n-i}a_{n-j} = \begin{cases} N, & i = j \\ -1, & i \neq j \end{cases}. \tag{9}$$

This matrix turns out to have a known inverse with ijth elements

$$\left(\sum_{n=1}^{N} a_{n-i}a_{n-j}\right)^{-1} = \begin{cases} \dfrac{2}{N+1}, & i = j \\ \dfrac{1}{(N+1)}, & i \neq j \end{cases} \tag{10}$$

and so (7) can be calculated to give

$$\sigma_H^2 = \sigma^2 \frac{2N}{N+1}.$$

Note that for orthogonal sequences the variance is σ^2, while with pseudorandom sequences the variance is enhanced by a factor of 2 when N is large. As we shall see next, the variance can be reduced by taking more than N samples in the composition of the matrix, Z_l.

III. THE FAST ALGORITHM

We saw in the previous section that the channel estimation problem entails the solution of linear equations. As was already mentioned, a desirable approach, from an implementation point of view, is to solve these equations recursively. To motivate the algorithm we return to the key equation, (3), and note that if $\hat{H}_{l-1}$ is the solution at time, "$l-1$" the solution at "l" can be calculated from

$$\hat{H}_l = \hat{H}_{l-1} + \frac{Z_{l-1}^{-1}A_l}{1 + A_l^t Z_{l-1}^{-1} A_l}(r_l - A_l^t \hat{H}_{l-1}). \tag{11}$$

This is a well known[1] and standard recursive procedure for solving (3), and it derives from the special property of the matrix Z_l, which we now discuss briefly.

To appreciate how (11) comes about, note that the matrix Z_l can be computed from Z_{l-1}

$$Z_l = Z_{l-1} + A_l A_l^t. \tag{12}$$

Likewise, the vector

$$U_l = \sum_{n=1}^{l} A_n r_n = U_{l-1} + A_l r_l. \tag{13}$$

Now, applying the matrix inversion lemma[2] to (12), and assuming that the inverses exist, we find that

$$Z_l^{-1} = Z_{l-1}^{-1} - \frac{Z_{l-1}^{-1} A_l A_l^t Z_{l-1}^{-1}}{1 + A_l^t Z_{l-1}^{-1} A_l}. \tag{14}$$

This is the key equation and, in conjunction with (13), makes it possible to claim (11). We note that the algorithm expressed in (13) is computationally complicated since it requires the calculation of a matrix recursion, (14), and multiplication of matrices by vectors at each iteration. For large N this becomes infeasible and simpler procedures are therefore sought. However, before proposing a simpler algorithm we need to review some properties of pseudorandom sequences.

The pseudorandom data sequences that are the inputs to the canceler during the start-up period derive from the binary sequences

$$X = x_1 x_2 x_3 x_4 \cdots$$

($x_i = 0$, or 1). The nth digit is computed from certain of the earlier digits by means of the recurrence relation

$$x_n = x_{n-c} + x_{n-b} \bmod 2,$$

where c and b are integers, $0 < c < b$. The actual data sequences $\{a_n\}$ that are applied to the canceler are the x_n's with "0" replaced by "−1".

Returning now to the sequence X, we remark that in spite of the fact that x_n is completely determined by the digits that precede it, the sequence X resembles in some respects a completely random sequence. The calculation of the sequence X is carried out in a shift register working in a closed loop and a mod 2 adder. It turned out that for special choices of c and b the sequence X is periodic with period $2^b - 1$.[2]

In our application, we will make use of the following known properties of the sequences

$$1)\ A_n^t A_m = \sum_{i=1}^{N} a_{n-i} a_{m-i} = \begin{cases} N, & n = m \\ -1, & n \neq m \end{cases}$$

$$2)\ Q^t A_n = 1, \quad \text{for} \quad n = 1, \cdots, N$$

$$Q = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}_1^N,$$

the all "1" vector

$$3)\ A_{n+N} = A_n, \text{ periodicity.}$$

Data sequences possessing properties 1 through 3 are also referred to as pseudorandom sequences. As a consequence of these properties, it is easy to verify the following:
Property (a):

$$Z_N = \sum_{n=1}^{N} A_n A_n^t$$

$$= (N+1)I - QQ^t, \ N \text{ x } N \text{ matrix},$$

where I is the identity matrix. This decomposition is possible since the ijth element of the matrix Z_N, $(Z_N)_{ij} = N$ for $i = j$ and -1 on the off-diagonals. This can readily be seen from property 1.
Property (b): The inverse matrix

$$Z_N^{-1} = \frac{1}{N+1}(I + QQ^t).$$

This can be derived from the matrix inversion lemma or verified by actually computing $Z_N Z_N^{-1} = I$.
Property (c): The set of vectors

$$B_n = Z_N^{-1} A_n, \ n = 1 \cdots N$$

are orthonormal to the vectors A_n, $n = 1, \cdots N$. This is a crucial property to what follows, so we prove that

$$A_m^t Z_N^{-1} A_n = \frac{A_m^t}{N+1}(I + QQ^t)A_n$$

$$= \frac{A_m^t A_n + (A_m^t Q)(Q^t A_n)}{N+1} \begin{matrix} = 0, \ n \neq m \\ = 1, \ n = m. \end{matrix}$$

These properties suggest an approach to a simple and rapidly converging tap-adjustment algorithm.

The key to the simplification of the algorithm, (11), is the recognition that Z_l^{-1} for $l < N$ does not exist, and so it is not possible to start the algorithm at $l = 1$. The basic idea is to replace Z_l by Z_N and thus obtain the simpler algorithm

$$\hat{H}_{n+1} = \hat{H}_n - Z_N^{-1} A_n e_n, \tag{15}$$

where the error at time n is again

$$e_n = A_n^t \hat{H}_n - r_n$$

$$= A_n^t(\hat{H}_n - H) - v_n.$$

Note that this is a measurable quantity at each iteration and $A_n e_n$ is just the gradient of the instantaneous squared error.

The algorithm expressed in (15) is remarkably simple since

$$Z_N^{-1} A_n = \frac{1}{N+1} (I + QQ^t) A_n$$

$$= \frac{1}{N+1} (A_n + Q),$$

which follows from the definition of Z_N and property (b). Inserting this into (15) we obtain

$$\hat{H}_{n+1} = \hat{H}_n - \frac{1}{N+1} (A_n + Q) e_n. \tag{16}$$

This form is immediately recognized as a slightly modified stochastic gradient algorithm with step size equal to $1/N + 1$ and the gradient vector, A_n, replaced by $A_n + Q$. It is nothing more than the original vector, A_n, in which a_n's equal to -1 are replaced by zero and $a_n = 1$ is replaced by $a_n = 2$. We wish to acknowledge that during the course of the development of this theory C. W. Farrow anticipated the form of this algorithm.

It now remains to demonstrate that (16) indeed converges "fast", by which we mean that it converges in N steps. Toward this end, define the error vector

$$\epsilon_n = \hat{H}_n - H,$$

and rewrite (16) in the form,

$$\epsilon_{n+1} = \epsilon_n - Z_N^{-1} A_n (A_n^t \epsilon_n - \nu_n)$$

$$= (I - B_n A_n^t) \epsilon_n + B_n \nu_n. \tag{17}$$

Iterating (17) yields explicitly

$$\epsilon_{n+1} = \prod_{k=1}^{n} (I - B_k A_k^t) \epsilon_1$$

$$+ \sum_{k=1}^{n} \prod_{j=k}^{n-1} (I - B_{j+1} A_{j+1}^t) B_k \nu_k. \tag{18}$$

This is the general solution but because of property (c), which states that $A_n^t B_{n-1} = 0$, we get a much simpler solution, which is the chief reason for the rapid convergence, namely,

$$\epsilon_{n+1} = \left(I - \sum_{k=1}^{n} B_k A_k^t\right) \epsilon_1 + \sum_{k=1}^{n} B_k \nu_k$$
$$= \left[I - Z_N^{-1}\left(\sum_{k=1}^{n} A_k A_k^t\right)\right] \epsilon_1 + Z_N^{-1}\left(\sum_{k=1}^{n} A_k \nu_k\right). \quad (19)$$

This simple form results because the product of the matrices in (18) reduces to

$$\prod_{k=1}^{n} (I - B_k A_k^t)\epsilon_1$$
$$= (I - B_1 A_1^t)(I - B_2 A_2^t) \cdots (I - B_n A_n^t)\epsilon_1$$
$$= [(I - B_1 A_1^t)(I - B_2 A_2^t) \cdots -B_n A_n^t]\epsilon_1$$
$$= \left[I - \sum_{k=1}^{n} B_k A_k^t\right] \epsilon_1.$$

The evolution of the error vector ϵ_n is guided by two components, the transient

$$\tau_n = \left[I - Z_N^{-1}\left(\sum_{k=1}^{n} A_k A_k^t\right)\right] \epsilon_1, \quad (20)$$

and the steady-state component

$$S_n = Z_N^{-1}\left(\sum_{k=1}^{n} A_k \nu_k\right). \quad (21)$$

A most crucial property of the transient solution is that at $n = N$, τ_n vanishes, and this is the reason for claiming "fast" convergence. Clearly, the transient solution cannot vanish before this time since the inverse doesn't even exist, and therefore we claim the algorithm convergence in the least possible number of steps. This most important property of the algorithm can be seen from (20), since

$$\tau_N = (I - Z_N^{-1} Z_N)\epsilon_1 = 0.$$

Consequently, the error vector at time $N + 1$ consist only of measurement noise, or the steady-state component

$$\epsilon_{N+1} = Z_N^{-1} \left(\sum_{k=1}^{N} A_k \nu_k \right). \tag{22}$$

We have thus demonstrated that the algorithm converges to the true solution in N steps since the transient vanishes at the end of the Nth iteration and, from that time on, the taps fluctuate around the true value due to measurement noise, ν_n, alone. The variance of the tap fluctuations can be calculated from the variance matrix

$$\begin{aligned} \rho_N &= E\{\epsilon_{N+1}\epsilon_{N+1}^t\} \\ &= Z_N^{-1} \left(\sum_{k,k'=1}^{N} A_k A_{k'}^t E\{\nu_k \nu_{k'}\} \right) Z_N^{-1} \\ &= \sigma^2 Z_N^{-1} = \frac{\sigma^2}{N+1} (I + QQ^t), \end{aligned} \tag{23}$$

where again we assumed that the ν_k's are identically and independently distributed. The error variance, σ_H^2, is therefore,

$$\begin{aligned} \sigma_H^2 &= \sigma^2 \text{ Trace } Z_N^{-1} \\ &= \frac{\sigma^2}{N+1} \text{Trace}[I + QQ^t] \\ &= \sigma^2 \frac{2N}{N+1}. \end{aligned} \tag{24}$$

This is precisely the value we obtained from solving the set of linear equations, (3), with pseudorandom input sequences. Thus, iterating N times provides the solution in a simple fashion.

It may turn out in applications that the value of variance obtained in N iterations is not sufficiently small. As seems reasonable, the noise variance can be reduced to any desired value by repeating the pseudorandom sequence of length N. To see this, consider a slightly modified tap-adjustment algorithm

$$H_{n+1} = H_n - \alpha Z_N^{-1} A_n e_n, \tag{25}$$

where now the scaler, α, is a fixed step size yet to be determined. Proceeding as before, the recursion for the tap error $\epsilon_n = H_n - H$ now becomes

$$\epsilon_{n+1} = (I - \alpha B_n A_n)\epsilon_n + \alpha B_n \nu_n, \tag{26}$$

with the concomitant solution

$$\epsilon_{n+1} = \left[I - \alpha Z_N^{-1}\left(\sum_{k=1}^{n} A_k A_k^t\right)\right]\epsilon_1 + \alpha Z_N^{-1}\left(\sum_{k=1}^{n} A_k \nu_k\right). \tag{27}$$

Again, examine the solution at $n = pN + 1$, where p is a positive integer, to obtain

$$\epsilon_{pN+1} = \left[I - \alpha Z_N^{-1}\left(\sum_{k=1}^{pN} A_k A_k^t\right)\right]\epsilon_1 + \alpha Z_N^{-1}\left(\sum_{k=1}^{pN} A_k \nu_k\right). \tag{28}$$

Since A_n is periodic with period N (property 3), we conclude that

$$\epsilon_{pN+1} = (1 - \alpha p)\epsilon_1 + \alpha Z_N^{-1}\left(\sum_{k=1}^{pN} A_k \nu_k\right), \tag{29}$$

and so we see that the transient component can be made to vanish when $\alpha = 1/p$. A straightforward calculation indicates that with this choice of α the variance is

$$\sigma_H^2 = \frac{2N}{p(N+1)}\,\sigma^2, \tag{30}$$

indicating a reduction by a factor of p—the number of times the pseudorandom sequence is repeated.

REFERENCES

1. M. S. Mueller, "Least-Squares Algorithms for Adaptive Equalizers," B.S.T.J., *60*, No. 8 (October 1981), pp. 1905–25.
2. B. D. O. Anderson and J. B. Moore, *Optimal Filtering*, Information and System Science Series, Thomas Kailath, ed., Englewood Cliffs, NJ: Prentice-Hall, 1979.
3. J. J. Stiffler, *Theory of Synchronous Communications*, Englewood Cliffs, NJ: Prentice-Hall, 1971, pp. 178–84.

AUTHOR

Jack Salz, B.S.E.E., 1955, M.S.E., 1956, and Ph.D., 1961, University of Florida; Bell Laboratories, 1961—. Mr. Salz first worked on the electronic switching system. Since 1968 he has supervised a group engaged in theoretical studies in data communications and is currently a member of the Communications Methods Research Department. During the academic year 1967–68, he was on leave as Professor of Electrical Engineering at the University of Florida. He was a visiting lecturer at Stanford University in Spring 1981 and a visiting MacKay Lecturer at the University of California, at Berkeley, in Spring 1983.

THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 6, July–August 1983
Printed in U.S.A.

Sample Reduction and Subsequent Adaptive Interpolation of Speech Signals

By R. STEELE* and F. BENJAMIN†

(Manuscript received December 15, 1982)

In this paper we investigate the effect of rejecting every nth speech sample and replacing it by means of adaptive interpolation. The interpolation procedure attempts to minimize the mean square interpolation error by recomputing the autocorrelation function of the speech sequence every W samples. We describe three methods of computing the correlation function. An iterative procedure is evaluated for estimating the correlation function of a speech sequence whose every nth sample has been discarded. For speech bandlimited to 3.2 kHz, sampled at 8 kHz, and $n = 4$, $W = 256$, the gain in signal-to-noise ratio (s/n) achieved by adaptive interpolation compared to nearest neighbor average interpolation was 14 and 8 dB, depending on whether the correlation function was computed from the original speech, or by using the iterative procedure, respectively. The effect of varying n from 2 to 6 was also investigated. Finally, we applied the interpolation procedures to 8-bit μ-law pulse code modulation (PCM), $\mu = 255$, reducing 64 kb/s transmission to 48 kb/s by rejecting one PCM word in four. The recovered speech after interpolation had a s/n that approximated that of conventional 56 kb/s μ-law PCM speech.

I. INTRODUCTION

Sampling of speech signals is usually performed at a rate high enough to prevent objectionable aliasing. Thus a speech signal whose bandwidth extends from 0.3 to 3.3 kHz is typically sampled at 8 kHz. After the speech signal has been sampled, the subsequent processing

* University of Southampton, Southampton, England. † Bell Laboratories.

of the samples depends on the application. In digital encoding the samples are converted into a digital (usually binary) sequence. Clearly, the lower the sampling rate the lower the bit rate of the digitally encoded speech, and the smaller the required channel capacity. There are numerous techniques for bit-rate reduction that have been compiled and categorized.[1,2] Our intention here is to dwell on a specific subset of bit-rate reduction, namely that arising from discarding speech samples at the transmitter or at a node in a network, and replacing them at the receiver by interpolation. The impetus for this approach arose from two unrelated interests, analog speech scramblers and bit-rate reduction in μ-law pulse code modulation (PCM). Scramblers that offer a reasonable amount of security, such as modulo masking scramblers, expand the bandwidth to half the sampling rate. Thus, operating on 8-kHz sampled speech, we wanted to reduce the sampling rate by discarding every fourth sample such that, after scrambling, the bandwidth would be 3 kHz, i.e., confined to the bandwidth of telephonic speech. The descrambled samples at the receiver would be brought back to the original 8-kHz rate by interpolating the missing samples. In the case of μ-law PCM our interest was to discard a percentage (typically 25 percent) of the words representing encoded speech, enabling other data to have a free ride. At the receiver the data would be removed and the missing speech samples recovered by interpolation.

Interpolation ideas are part of everyday life; we are always filling in the gaps in our perception by interpolation, extrapolation, and prediction. We will avoid the luxury of philosophizing, and briefly review some of the interesting aspects of interpolation that are related to speech encoding. Mathews,[3] in an attempt to make significant reductions in the sampling rate of speech, considered extremal encoding. With this technique the amplitudes of the speech signal at its extremes and the time intervals between these extremes are transmitted in an encoded format. At the receiver the extremes can be widely separated, and the interpolation of the missing samples can be thought of as curve fitting, i.e., passing a quadratic function through the extreme sample and two nearest received samples. In the 1960's there was considerable interest in interpolation, which Kortman[4] defined as "the process of after-the-fact polynomial curve fitting to eliminate redundant data samples." The interpolators[4–6] tended to be concerned with zero-order, first-order, and fan interpolators, and were closely related to aperture predictors.[4–7] Andrews et al.[6] considered straight-line optimum, optimum interpolation filter, $(\sin x)/x$, Lagrange, and Fourier reconstruction interpolators. The use of piecewise polynomials, called splines, have been extensively investigated (see Ref. 8 and its bibliography). More recently, there have been a spate of publications[9] on

digital speech interpolation, although the interpolation there was mainly concerned with Time Assignment Speech Interpolation (TASI)-type systems. Interpolation techniques have also been applied to reduce distortion in packet switching of speech when a packet is lost or discarded.[10] For an understanding of the existing low-pass filtering procedures of interpolation and decimation of digital signals, the reader is directed to the in-depth review presented by Crochiere and Rabiner.[11]

Having commented on some of the existing interpolation methods let us now proceed to the issues to be addressed here. Consider the situation of being presented with the speech samples, or say μ-law PCM words, at some point in a communications network, together with a system control demand to eliminate J samples (or words) per W samples (or words). This could happen, for example, if there were a sudden increase in traffic. Rejection of these J samples might precipitate unacceptable degradation in the recovered speech unless we introduced at the receiver replacement samples that closely approximate those samples rejected. We must therefore decide at the outset on the means of reinserting the J samples, for example, by prediction or interpolation, using those samples not discarded. Having replaced the missing samples, we need to establish criteria for judging the quality of the recovered speech signal. Is a single criterion such as mean square error sufficient, or is a combination of objective and subjective measures required? Coupled with the issues of reinsertion and quality are the criteria of how to select which J samples are to be rejected in every W samples. Should we determine if speech is present as distinct from silence, and if so, whether it is voiced or unvoiced? Is it better to reject small groups of samples (e.g., in silence periods), leaving clusters of samples intact because rejection followed by subsequent interpolation of one of these samples might cause significant speech degradation, or should we always endeavor to retain samples on either side of a rejected sample, and so on?

Clearly, the choices are legion, and faced with this situation we have imposed a set of guidelines that are not concerned with what the samples (or words) represent in the speech signal. For example, we make no attempt to recognize if the talker is male or female, whether the speech is voiced or unvoiced, or in transition, and so on. Instead, we simply reject samples on a periodic basis, and reinsert them at their destination by means of adaptive interpolation. By making the interpolation adaptive we take cognizance of the local statistics of the speech signal, specifically, the utilization of the signal's correlative properties. This is the only characteristic of the speech signal that is exploited. Our approach is, therefore, oriented to ease implementation rather than to squeeze the maximum advantage from the properties

of speech. Indeed, the method is applicable to other types of analog signals (e.g., modem-generated data signals), provided the signals possess correlative properties that are capable of exploitation in the interpolation processes. Our strategy is as follows:

1. Discard samples on a periodic basis, with no two consecutive samples being rejected.
2. Process blocks of W samples at a time.
3. Exploit the local statistics of the speech samples by determining their correlation function over each block. Approximate methods for determining the correlation function are required when the sequence has had J of its W samples rejected.
4. Apply adaptive interpolation.
5. Employ quality criteria based on the minimization of the mean square error, justified by the prior knowledge that the mean square error values achieved will conform to perceptual standards based on informal listening tests.

Having introduced the notion of sample or word rejection to reduce the baud or bit rate, and an outline of how to discard the samples or words and reintroduce them at their destination, we will now formulate the problem and its solution in detail.

II. THE PROBLEM

Consider a speech signal $x(t)$, bandlimited to $f_c H_z$ and sampled at $f_s H_z$ to yield the sequence $\{x_k\}$, where f_s satisfies

$$f_s \geq 2f_c. \tag{1}$$

This sequence $\{x_k\}$ could be encoded into binary words, or we could be given binary words at a node in the network. In such situations our description of the problem would be in data words. However, for ease of explanation we will confine our discussion to operations on samples unless otherwise stated.

To reduce the number of samples per second in $\{x_k\}$, we discard every nth sample. This is achieved by clocking the speech samples into a gear-down changing buffer under the auspices of a clock operating at f_s samples per second, such that, after every n samples are inserted into the buffer, the clock is inhibited for one clock period. The arrangement is shown in Fig. 1. The samples are clocked out of the buffer at a rate

$$F_s < 2f_c \tag{2}$$

to yield a sequence $\{y_k\}$ whose components are uniformly spaced in time by $1/F_s$. Figure 2 shows $\{x_k\}$ and $\{y_k\}$ for arbitrary segments of sampled speech. Observe that $\{y_k\}$ has its parameter F_s related to f_s by

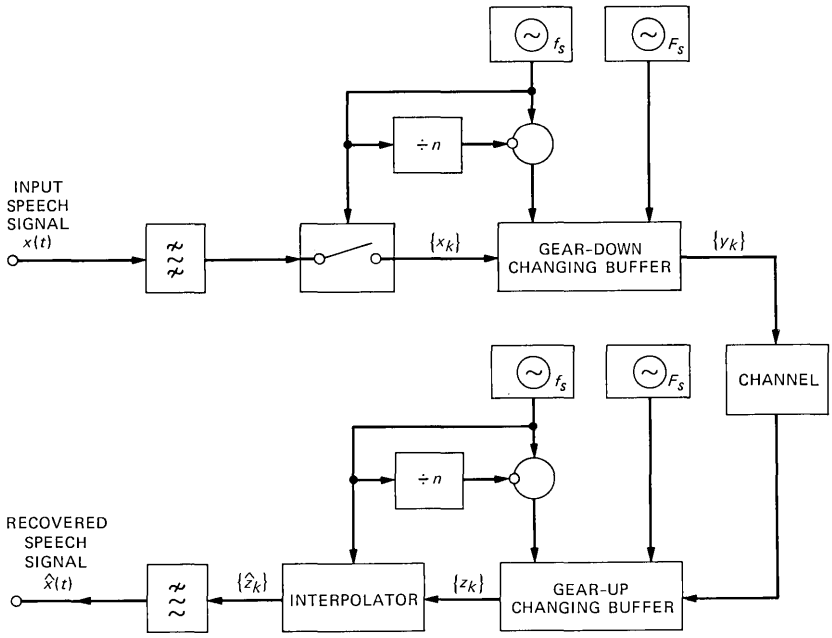


Fig. 1—Arrangement for decreasing the sample rate.

$$F_s = \left(\frac{n-1}{n}\right) f_s. \tag{3}$$

Suppose $\{y_k\}$ with its symbol rate lower than $\{x_k\}$ is transmitted over an ideal channel. Let us further assume that the receiver is able to return the samples in $\{y_k\}$ to the relative time positions they occupied in the original speech sequence $\{x_k\}$ using the gear-up changing buffer shown in Fig. 1. This newly formed sequence, $\{z_k\}$, generated at a rate $1/f_s$, has one out of every n samples absent owing to the sample reduction process at the transmitter. A small, arbitrary segment of this sequence $\{z_k\}$, corresponding to a particular $\{x_k\}$ and $\{y_k\}$, is displayed in Fig. 2. Our problem is to replace those samples rejected at the transmitter with substitutes of acceptable accuracy whose generation is not excessively complex. The approach employed is adaptive interpolation.

III. ADAPTIVE INTERPOLATION

The speech sequence $\{z_k\}$ is divided into sequential blocks having W sample positions spaced $1/f_s$ seconds apart. The sequence $\{z_k\}$ is therefore composed of W samples from $\{x_k\}$ with every nth sample absent. Our purpose is to interpolate the missing samples to produce

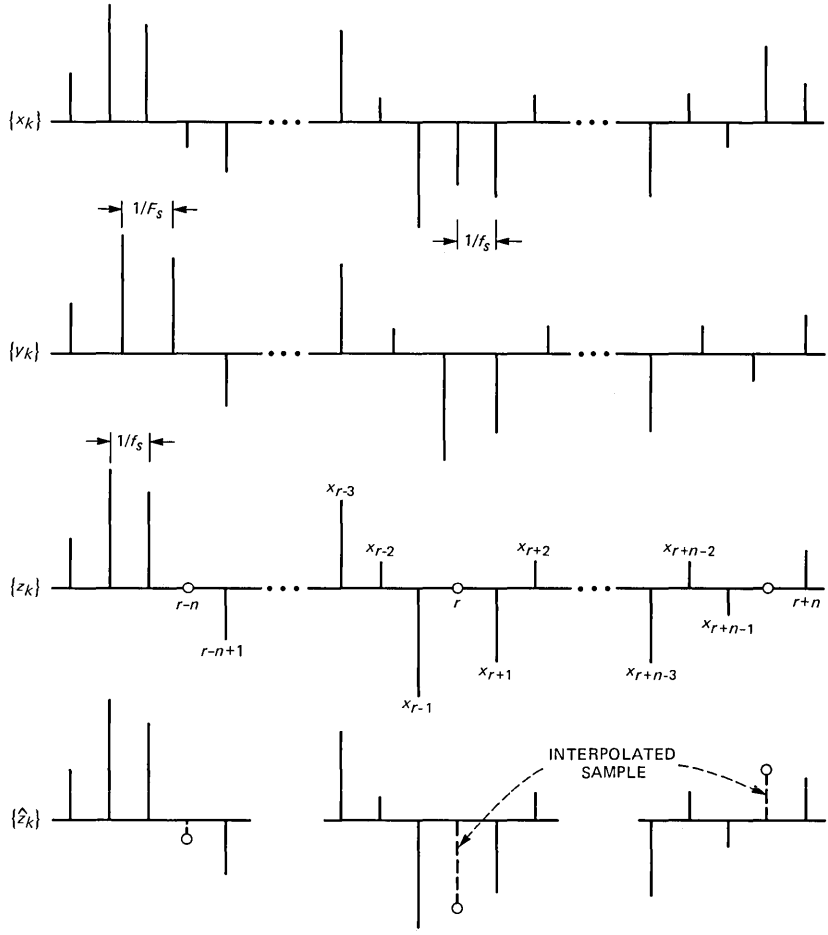


Fig. 2—Sampled sequences for arbitrary segment of speech; $\{x_k\}$ is the original speech sequence; $\{y_k\}$ is the sequence after sample reduction and gearing changing; $\{z\}$ is $\{x_k\}$ with every nth sample absent; $\{\hat{z}\}$ is the recovered speech sequence.

a sequence for the first block having components

$$x_1, \cdots, x_{n-1}, \hat{z}_n, x_{n+1}, \cdots, x_{W-2}, x_{W-1}\hat{z}_W,$$

where

$$\hat{z}_r; \qquad r = n, 2n, \cdots, W - n, W$$

are the interpolated samples. When interpolating each speech sample, we will use λ past and λ previous samples, and restrict λ to be

$$\lambda \leq n - 1. \tag{4}$$

Thus, a missing sample at the rth sampling instant (see Fig. 2) is

formed by interpolation according to

$$\hat{z}_r = \sum_{i=-\lambda}^{-1} a_i x_{r+i} + \sum_{i=1}^{\lambda} a_i x_{r+i}, \tag{5}$$

where a_i are the interpolation parameters. Equation (5) may be written as

$$\hat{z}_r = \sum_{i=-\lambda}^{\lambda} a_i x_{r+i}, \tag{6}$$

where $a_0 = 0$. The interpolation error of the rth sample is

$$e_r = x_r - \hat{z}_r \tag{7}$$

and the square of this error is

$$\begin{aligned} e_r^2 &= x_r^2 - 2x_r \sum_{i=-\lambda}^{\lambda} a_i x_{r+i} + \left(\sum_{i=-\lambda}^{\lambda} a_i x_{r+i} \right)^2 \\ &= x_r^2 - 2x_r \sum_{i=-\lambda}^{\lambda} a_i x_{r+i} + \sum_{i=-\lambda}^{\lambda} a_i^2 x_{r+i}^2 + 2 \sum_{i=-\lambda}^{\lambda} \sum_{j=-\lambda}^{\lambda} a_i x_{r+i} a_j x_{r+j} \end{aligned} \tag{8}$$

with $i \neq j$ and $j > i$. To determine the interpolation parameters we proceed as follows. The square of the error is summed over the block of samples

$$\begin{aligned} E_r^2 = \sum_{r=n}^{W} e_r^2 = \sum_{r=n}^{W} x_r^2 - 2 \sum_{r=n}^{W} \left\{ x_r \sum_{i=-\lambda}^{\lambda} a_i x_{r+i} \right\} \\ + \sum_{r=n}^{W} \sum_{i=-\lambda}^{\lambda} a_i^2 x_{r+i}^2 + 2 \sum_{r=n}^{W} \sum_{i=-\lambda}^{\lambda} \sum_{j=-\lambda}^{\lambda} a_i x_{r+i} a_j x_{r+j}, \end{aligned} \tag{9}$$

where $r = n, 2n, \cdots, W$. To select coefficients that minimize this summation, we partially differentiate E_r^2 with respect to each of the coefficients and set the result to zero. After rearranging the equations we have

$$\sum_{r=n}^{W} x_r x_{r+j} = \sum_{i=-\lambda}^{\lambda} \sum_{r=n}^{W} a_i x_{r+i} x_{r+j}, \tag{10}$$

where

$$\begin{aligned} j &= \pm 1, \pm 2, \cdots, \pm\lambda \\ i &= -\lambda, -\lambda + 1, \cdots, \lambda \\ r &= n, 2n, \cdots, W - n, W \\ W &\geq 2n. \end{aligned}$$

The interpolation coefficients in eq. (10) can be represented in vector form as

$$\alpha = A^{-1}C, \tag{11}$$

where

$$\alpha = [a_{-\lambda}, a_{-\lambda+1}, \cdots, a_{-1}, a_1, \cdots, a_{\lambda-1}, a_\lambda]^T, \tag{12}$$

$$C = [R(0, -\lambda), R(0, -\lambda + 1), \cdots, R(0, -1), R(0, 1), \cdots, R(0, \lambda - 1), R(0, \lambda)]^T, \tag{13}$$

and the superscripts -1 and T represent inverse and transpose operations, respectively. The rectangular matrix A is of order 2λ, and its elements are presented in Table I. The concentric dotted enclosures, commencing with the inner one, refer to the A matrix for $\lambda = 1, 2, 3, \cdots$. The elements are given by

$$R(k, j) = \frac{\sum_{r=n}^{W} x_{r+k}x_{r+j}}{\sum_{r=n}^{W} x_{r+j}^2}$$

$$k = \pm 1, \pm 2, \cdots, \pm\lambda$$

$$j = \pm 1, \pm 2, \cdots, \pm\lambda$$

$$r = n, 2n, \cdots, W, \tag{14}$$

and $R(k, j)$ will be referred to as the correlation function $R(k, j)$. The vector C has elements $R(0, j)$, i.e., the elements are given by eq. (13) with k always zero.

As an example of the application of eq. (11), consider the case of $W = 32$, $n = 4$, $\lambda = 2$, whence

$$\begin{bmatrix} a_{-2} \\ a_{-1} \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} 1 & R(-1, -2) & R(1, -2) & R(2, -2) \\ R(-2, -1) & 1 & R(1, -1) & R(2, -1) \\ R(-2, 1) & R(-1, 1) & 1 & R(2, 1) \\ R(-2, 2) & R(-1, 2) & R(1, 2) & 1 \end{bmatrix}^{-1} \begin{bmatrix} R(0, -2) \\ R(0, -1) \\ R(0, 1) \\ R(0, 2) \end{bmatrix} \tag{15}$$

and eq. (14) becomes for the A and C matrices

$$R(k, j) = \frac{\sum_{r=4}^{32} x_{r+k}x_{r+j}}{\sum_{r=4}^{32} x_{r+j}^2}; \quad \begin{array}{l} k = 0, \pm 1, \pm 2 \\ j = \pm 1, \pm 2 \\ r = 4, 8, 12, 16, 20, 24, 28, 32 \end{array}$$

Table I—The A matrix

1	$R(-\lambda+1, -\lambda)$	· ·	$R(-5, -\lambda)$	$R(-4, -\lambda)$	$R(-3, -\lambda)$	$R(-2, -\lambda)$	$R(-1, -\lambda)$	$R(1, -\lambda)$	$R(2, -\lambda)$	$R(3, -\lambda)$	$R(4, -\lambda)$	$R(5, -\lambda)$	· ·	$R(\lambda-1, -\lambda)$	$R(\lambda, -\lambda)$
$R(-\lambda, -\lambda, +1)$	· ·	· ·	·	·	·	·	·	·	·	·	·	·	· ·	· ·	$R(\lambda, -\lambda+1)$
·	· ·	· ·	·	·	·	·	·	·	·	·	·	·	· ·	· ·	·
·	· ·	· ·	·	·	·	·	·	·	·	·	·	·	· ·	· ·	·
$R(-\lambda, -5)$	· ·	· ·	1	$R(-4, -5)$	$R(-3, -5)$	$R(-2, -5)$	$R(-1, -5)$	$R(1, -5)$	$R(2, -5)$	$R(3, -5)$	$R(4, -5)$	$R(5, -5)$	· ·	· ·	$R(\lambda, -5)$
$R(-\lambda, -4)$	· ·	· ·	$R(-5, -4)$	1	$R(-3, -4)$	$R(-2, -4)$	$R(-1, -4)$	$R(1, -4)$	$R(2, -4)$	$R(3, -4)$	$R(4, -4)$	$R(5, -4)$	· ·	· ·	$R(\lambda, -4)$
$R(-\lambda, -3)$	· ·	· ·	$R(-5, -3)$	$R(-4, -3)$	1	$R(-2, -3)$	$R(-1, -3)$	$R(1, -3)$	$R(2, -3)$	$R(3, -3)$	$R(4, -3)$	$R(5, -3)$	· ·	· ·	$R(\lambda, -3)$
$R(-\lambda, -2)$	· ·	· ·	$R(-5, -2)$	$R(-4, -2)$	$R(-3, -2)$	1	$R(-1, -2)$	$R(1, -2)$	$R(2, -2)$	$R(3, -2)$	$R(4, -2)$	$R(5, -2)$	· ·	· ·	$R(\lambda, -2)$
$R(-\lambda, -1)$	· ·	· ·	$R(-5, -1)$	$R(-4, -1)$	$R(-3, -1)$	$R(-2, -1)$	1	$R(1, -1)$	$R(2, -1)$	$R(3, -1)$	$R(4, -1)$	$R(5, -1)$	· ·	· ·	$R(\lambda, -1)$
$R(-\lambda, 1)$	· ·	· ·	$R(-5, 1)$	$R(-4, 1)$	$R(-3, 1)$	$R(-2, 1)$	$R(-1, 1)$	1	$R(2, 1)$	$R(3, 1)$	$R(4, 1)$	$R(5, 1)$	· ·	· ·	$R(\lambda, 1)$
$R(-\lambda, 2)$	· ·	· ·	$R(-5, 2)$	$R(-4, 2)$	$R(-3, 2)$	$R(-2, 2)$	$R(-1, 2)$	$R(1, 2)$	1	$R(3, 2)$	$R(4, 2)$	$R(5, 2)$	· ·	· ·	$R(\lambda, 2)$
$R(-\lambda, 3)$	· ·	· ·	$R(-5, 3)$	$R(-4, 3)$	$R(-3, 3)$	$R(-2, 3)$	$R(-1, 3)$	$R(1, 3)$	$R(2, 3)$	1	$R(4, 3)$	$R(5, 3)$	· ·	· ·	$R(\lambda, 3)$
$R(-\lambda, 4)$	· ·	· ·	$R(-5, 4)$	$R(-4, 4)$	$R(-3, 4)$	$R(-2, 4)$	$R(-1, 4)$	$R(1, 4)$	$R(2, 4)$	$R(3, 4)$	1	$R(5, 4)$	· ·	· ·	$R(\lambda, 4)$
$R(-\lambda, 5)$	· ·	· ·	$R(-5, 5)$	$R(-4, 5)$	$R(-3, 5)$	$R(-2, 5)$	$R(-1, 5)$	$R(1, 5)$	$R(2, 5)$	$R(3, 5)$	$R(4, 5)$	1	· ·	· ·	$R(\lambda, 5)$
·	· ·	· ·	·	·	·	·	·	·	·	·	·	·	· ·	· ·	·
·	· ·	· ·	·	·	·	·	·	·	·	·	·	·	· ·	· ·	·
	· ·	· ·	·	·	·	·	·	·	·	·	·	·	· ·	· ·	
$R(-\lambda, \lambda-1)$	· ·	· ·	·	·	·	·	·	·	·	·	·	·	· ·	· ·	$R(\lambda, \lambda-1)$
$R(-\lambda, \lambda)$	$R(-\lambda+1, \lambda)$	· ·	$R(-5, \lambda)$	$R(-4, \lambda)$	$R(-3, \lambda)$	$R(-2, \lambda)$	$R(-1, \lambda)$	$R(1, \lambda)$	$R(2, \lambda)$	$R(3, \lambda)$	$R(4, \lambda)$	$R(5, \lambda)$	· ·	$R(\lambda-1, \lambda)$	1

When $r = 32$ and $|k|$ or $|j| \geq 1$, samples are used that reside in the subsequent block.

When the block size W is large, typically in excess of 256 samples, the computations required to solve eq. (11) can be reduced at the expense of a small increase in interpolation noise power. This is accomplished by replacing $R(k,j)$ of eq. (14) by the correlation function

$$R(\tau) = \frac{(W-\tau)\sum_{i=1}^{W-\tau} x_i x_{i+\tau} - \left(\sum_{i=1}^{W-\tau} x_i\right)\left(\sum_{i=1}^{W-\tau} x_{i+\tau}\right)}{\left\{\left[(W-\tau)\sum_{i=1}^{W-\tau} x_i^2 - \left(\sum_{i=1}^{W-\tau} x_i\right)^2\right] \cdot \left[(W-\tau)\sum_{i=1}^{W-\tau} x_{i+\tau}^2 - \left(\sum_{i=1}^{W-\tau} x_{i+\tau}\right)^2\right]\right\}^{1/2}}, \tag{16}$$

where

$$\tau = |k-j| \tag{17}$$

and k and j have values prescribed in eq. (14). For a further small increase in interpolation noise, which typically reduces the recovered signal-to-noise ratio (s/n) by a couple of decibels compared to when eq. (16) is used, $R(\tau)$ can be simplified to

$$R(\tau) = \frac{\sum_{r=1}^{W-\tau} x_r x_{r+\tau}}{\sum_{r=1}^{W} x_r^2}, \tag{18}$$

where r is integer-valued. Observe that, in computing the interpolation coefficients with the aid of eqs. (16) or (18), only λ values need be determined as

$$a_{-p} = a_p; \qquad p = 1, 2, \cdots, \lambda. \tag{19}$$

By contrast, when $R(k,j)$ is used in preference to $R(\tau)$, eq. (19) does not apply and 2λ coefficients must be computed.

Thus, provided we can compute $R(k,j)$ or $R(\tau)$, over a duration of W/f_s, where W is the block length, we can determine the interpolation parameters contained in the vector α. Employing eq. (5) we can estimate the missing samples by means of interpolation. The mean squared error between $\{x_k\}$, and the recovered sequence $\{\hat{z}_k\}$ containing the interpolated samples (see the example shown in Fig. 2), is approximately minimized, provided W is sufficiently large. Practical values of W are determined in Section VI.

IV. INTERPOLATION PARAMETERS DERIVED FROM THE INPUT DATA

The interpolation parameters are computed by first finding the correlation function $R(\theta)$, where θ is k, j, or τ. Equation (14) shows $R(\theta)$ as dependent on the input sequence $\{x_k\}$. Clearly, as $\{x_k\}$ is only known at the transmitter, it follows that the interpolation vector α must be computed at the transmitter and multiplexed with the slowed-down speech samples $\{y_k\}$. Consequently,

$$Y = \left(\frac{n-1}{n}\right) W + \nu \tag{20}$$

samples are transmitted every W/f_s seconds, where Y is composed of $W(n-1)/n$ speech samples and ν interpolation parameter samples. The value of ν is λ or 2λ, depending on whether θ is τ or k, j, respectively. This means that F_s of eq. (3) is modified to

$$F_s' = F_s + \frac{\nu}{W} f_s. \tag{21}$$

For example, if $f_s = 8$ kHz, $n = 4$, $F_s = 6$ kHz, $W = 256$, $\lambda = 3$, $\theta = \tau$, then $F_s' = 6.093$ kHz. We also observe from eq. (20) that three interpolation samples are sent for every 192 speech samples. The values of λ and W as a function of s/n are present in Section VI.

V. INTERPOLATION PARAMETERS DERIVED FROM THE RECEIVED DATA

The receiver produces the sequence $\{z_k\}$ whose samples are spaced apart by $1/f_s$, and in every nth sample position one sample is missing, as shown in Fig. 2. The receiver has the task of estimating the interpolation coefficients from $\{z_k\}$, and must therefore commence by calculating the correlation function $R(\tau)$ without the full knowledge of the original speech sequence $\{x_k\}$. In this situation we proceed as follows. The missing samples are found as an average of adjacent speech samples

$$\bar{x}_r = (x_{r-1} + x_{r+1})/2 \tag{22}$$

until a sequence $\{\bar{z}_k\}$ consisting of $W(n-1)/n$ original speech samples and W/n interpolated samples is formed, where each sample is equally spaced from its neighbor by $1/f_s$ seconds. This sequence corresponds to a recovered speech signal that has considerable distortion, particularly for female speakers. Instead of accepting $\{\bar{z}_k\}$ as the recovered speech sequence, we use it solely for the purpose of computing $R(\tau)$, and hence the vector α containing a_{-1} and a_1. We now remove the samples introduced by eq. (22) and replace them by using adaptive interpolation based on the two nearest neighbors, viz:

$$ {}_2x_r = a_{-1}x_{r-1} + a_1 x_{r+1}, \tag{23} $$

where a_{-1} and a_1 are found using eq. (11) with $R(k, j)$ replaced by $R(\tau)$ of eq. (18). The new recovered speech sequence $\{{}_2z_k\}$ has interpolated samples formed according to eq. (23), and in general contains less distortion than $\{\bar{z}_k\}$. However, we can further reduce the distortion. The function $R(\tau)$ is again computed, this time from $\{{}_2z_k\}$. The interpolated samples in $\{{}_2z_k\}$, derived with the aid of eq. (23), are rejected and replaced by

$$ {}_4x_r = a_{-2}x_{r-2} + a_{-1}x_{r-1} + a_1 x_{r+1} + a x_{r+2} \tag{24} $$

to yield the speech sequence $\{{}_4z_k\}$. The correlation function $R(\tau)$ of sequence $\{{}_4z_k\}$ is found, and those interpolated samples previously formulated with the aid of eq. (24) are exchanged for

$$ {}_6x_r = \sum_{i=-3}^{3} a_i x_{r-i}, \qquad a_0 = 0. \tag{25} $$

This process of using two more samples per iteration in the interpolation procedure continues until the limits of the summation in eq. (25) become $n - 1$, when the final recovered speech sequence $\{\hat{z}_k\}$ is obtained. It should be noted that more than $2(n - 1)$ samples can be used in each interpolation process, but the improvement in interpolation accuracy results in a significant increase in complexity.

Often it is sufficient to produce the sequence $\{\bar{z}_k\}$, compute $R(\tau)$ from $\{\bar{z}_k\}$, remove the samples formed by averaging the adjacent two samples, and with the aid of $\lambda \leq n - 1$ samples on either side of each discarded sample, insert the missing samples by adaptive interpolation. When this process is adopted, it will be referred to as λ-interpolation. However, when the same λ samples are used in the interpolation process, and the iteration procedure of eqs. (22) to (25) activated, we will refer to this interpolation scheme as λ-with-iteration.

Observe that $R(\tau)$ is used in the λ-with-iteration scheme. If $R(k, j)$ is employed instead of $R(\tau)$, the final interpolation is not better than that of the sample used to approximate x_r. For example, if we replaced x_r by $\bar{x}_r$, the iterative algorithm would merely attempt to minimize the error power between the $\bar{x}_r$ samples and the interpolated samples. Unlike the application of $R(k, j)$, the $R(\tau)$ function does not enable an exact minimization of interpolation error power to be achieved at the transmitter. However, it transpires that by computing $R(\tau)$ at the receiver, and using the iterative procedure to interpolate the missing samples, the interpolation noise is significantly reduced (see Section 6.2).

VI. RESULTS

The speech signal used in our experiments consisted of two con-

catenated sentences, "Live wires should be kept covered," and "To reach the end he needs much courage." These were spoken by a male and female, respectively. The signal was bandlimited from 0.3 to 3.2 kHz and sampled at 8 kHz to give the input speech sequence $\{x_k\}$. This sequence is displayed in Fig. 3, together with its spectrogram, where the higher frequencies have been preemphasized.

6.1 Interpolation parameters generated from the original speech

The gear-down changing procedure was invoked (see Figs. 1 and 2) whereby the sampling rate of 8 kHz was decreased to a uniform F_s-kHz rate by rejecting every nth sample, and adjusting the sample spacing to provide the output sequence $\{y_k\}$. This sequence was assumed to be transmitted through an ideal channel, and after passing through the receiver's gear-up changing buffer the sequence $\{z_k\}$ was formed. The sequence $\{z_k\}$ had a symbol rate of 8 kHz, with every nth sample absent. The absentee samples rejected at the transmitter were then formulated according to eq. (6). The interpolation parameters a_i were found with the aid of eq. (11), which used correlation functions $R(k, j)$ related to the input speech sequence $\{x_k\}$. Thus, in our first experiment we were concerned with how successfully we could discard every nth sample in $\{x_k\}$, and replace the discarded samples using interpolation parameters based on $\{x_k\}$.

We used as a performance criterion segmental signal-to-noise ratio[12] (SEG-s/n), computed using the input sequence $\{x_k\}$, and the error sequence $\{e_k\}$ whose components are given by eq. (7). In our initial experiment the variation of SEG-s/n as a function of block size W was found using practical values of W extending from 64 to 1024, and $n = 4$. This value of n is a compromise between providing adequate SEG-s/n and a reasonable reduction in the number of transmitted samples. Applying the correlation function $R(k, j)$ in the determination of the interpolation parameters, we obtained the solid curves in Fig. 4. Curves a, b, and c apply for the case of $\lambda = 3$, 2, and 1, respectively. Curve d is the SEG-s/n when the interpolation was performed using the average of adjacent samples. Increasing λ to $n - 1$, i.e., to 3, resulted in an increase in SEG-s/n compared to lower values of λ, and for a block size of 256 and $\lambda = 3$, a gain of 13.7 dB in SEG-s/n was obtained compared to the simple interpolation averaging method of eq. (22). When the correlation function $R(\tau)$ of eq. (18) was employed to compute the interpolation parameters, curves e, f, and g were obtained corresponding to $\lambda = 3$, 2, and 1, respectively. The improvement in SEG-s/n derived from employing $R(k, j)$ rather than $R(\tau)$ increased with increasing λ, being 0.1, 1, and 3 dB for $\lambda = 1$, 2, and 3, respectively, and with $W = 256$.

As a means of providing further insight into the performance of the

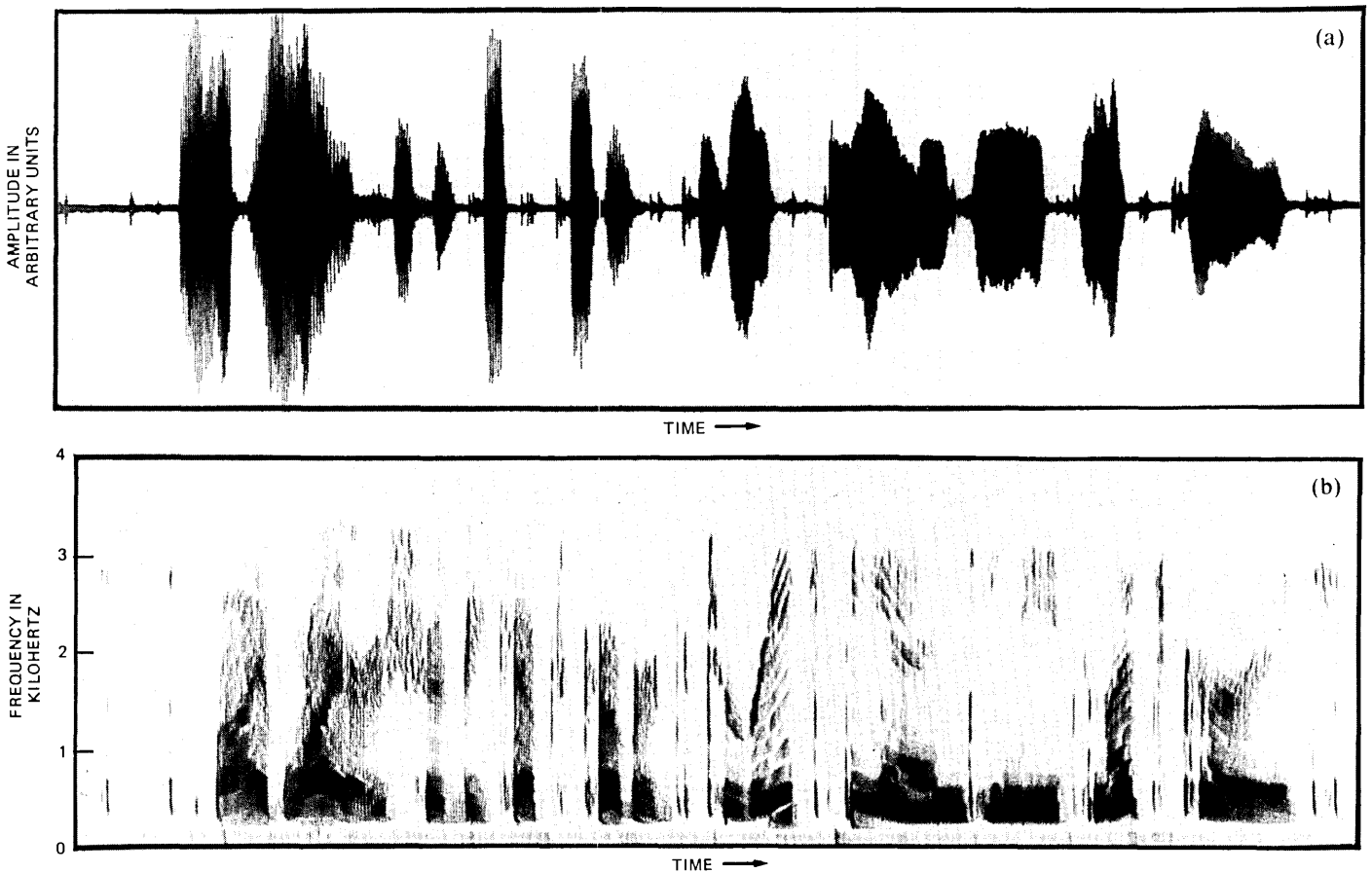


Fig. 3—(a) Input speech sequence $\{x_k\}$ and (b) its preemphasized spectrogram.

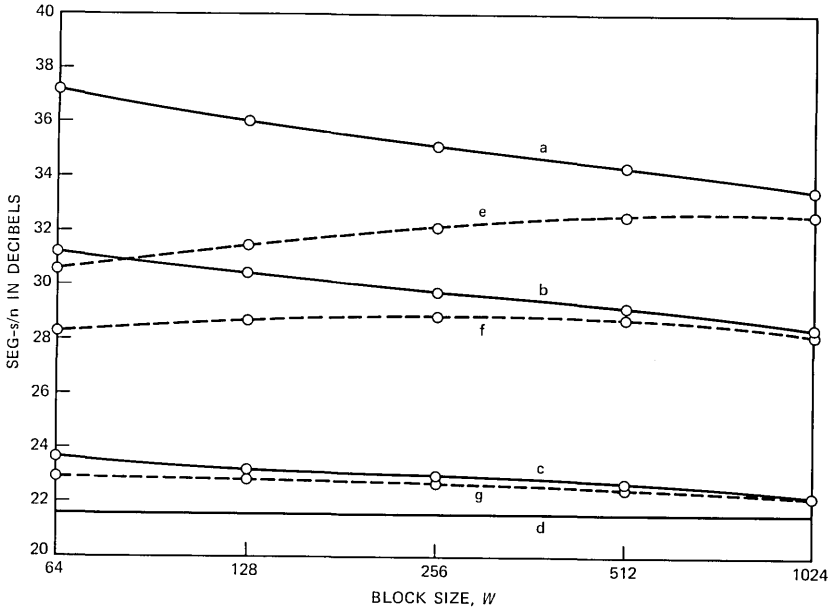


Fig. 4—SEG-s/n versus block size W. Curves a, b, and c apply for $\lambda = 3$, 2, and 1; and $\hat{R}(k, j)$ was used in calculating the interpolation parameters. Curve d relates to interpolation using nearest neighbor averaging. Curves e, f, and g apply for $\lambda = 3$, 2, and 1; and $R(\tau)$, given by eq. (18), was used in calculating the interpolation parameters.

interpolation system, we show in Fig. 5 the variation of the s/n of each block in the speech signal as a function of successive blocks for $W = 256$, $n = 4$. The average of these s/n values constitutes the SEG-s/n points in Fig. 4 for $W = 256$. As expected, we found that the s/n in every block was greater, if only by an infinitesimal amount, when more samples were used in the interpolation process, i.e., when larger values of λ were employed. Interpolation by adjacent sample averaging always provided the lowest s/n. In some blocks the advantage of using $\lambda = 3$ compared to $\lambda = 2$, $\lambda = 1$, and nearest neighbor averaging, provided s/n gains as large as 13, 31, and 33 dB, respectively.

Returning to Fig. 4, the SEG-s/n of over 35 dB, $\lambda = 3$, was found to be approximately 3 dB greater than the coventional s/n computed by measuring the mean square values of the components in $\{x_k\}$ and $\{e_k\}$ over the entire speech input signal. Our SEG-s/n measurements therefore indicate that the recovered speech is similar to toll quality speech.[2] Informal listening experiences for the recovered speech sequence $\{\hat{z}_k\}$ when $\lambda = 3$, $W = 256$, tended to confirm the SEG-s/n findings that the quality of the recovered speech sequence $\{\hat{z}_k\}$ was judged to be very similar to that of the original bandlimited sequence $\{x_k\}$. We observed that although the distortion in $\{\hat{z}_k\}$ for $\lambda = 1$ was

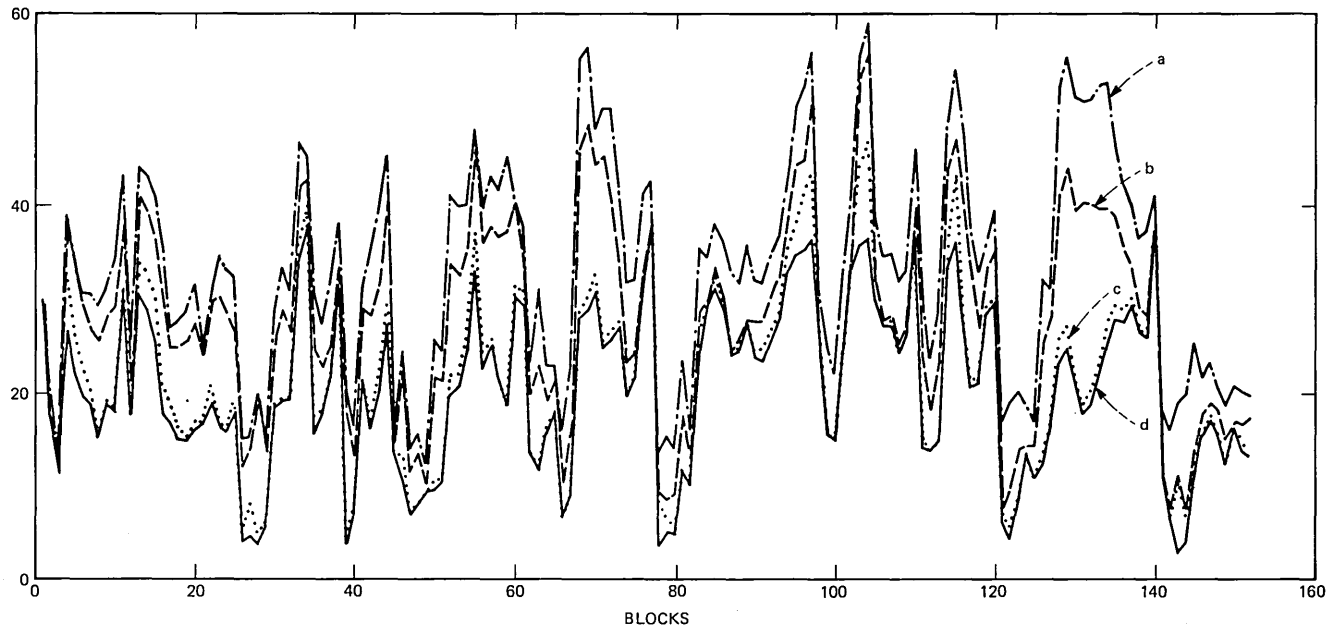


Fig. 5—Variation of block s/n as a function of block number, $W = 256$, $n = 4$. Interpolation procedure used (a) $\lambda = 3$, (b) $\lambda = 2$, (c) $\lambda = 1$, and (d) nearest neighbor averaging.

annoying, by making $\lambda = 2$ the interpolation noise was barely noticeable.

In Section III we present different expressions for the correlation function $R(\theta)$, namely eq. (14), where $\theta = k, j$, and eqs. (16) or (18) having $\theta = \tau$. The SEG-s/n and block s/n values shown in Figs. 4 and 5 were determined using $\theta = k, j$. We now demonstrate the loss in SEG-s/n due to using $\theta = \tau$ compared to when $\theta = k, j$ as a function of W, for the conditions of $n = 4$ and $\lambda = 3$. It will be recalled that $\theta = k, j$ enables the interpolation samples to be formulated that minimize E_r^2 over a block of W samples. When $\theta = \tau$ a low but not minimum value of E_r^2 is produced over the working range of W. The application of eq. (16) gives a more accurate measure of $R(\tau)$ than the simpler expression of eq. (18). Figure 6 shows the variation of SEG-s/n with block size when $R(\theta)$ is computed using eqs. (14), (16), and (18), $n = 4$, $\lambda = 3$. When $R(k, j)$ was used, the effect of increasing W was to decrease the SEG-s/n. This is to be expected because the interpolation parameters are fixed for a block, and we may think of a large block as composed of many smaller blocks, each with its optimum interpolation parameters. Thus, if one set of parameters is selected for the large block, these parameters are inevitably suboptimum for the smaller subblocks, and hence the SEG-s/n is lower for the larger

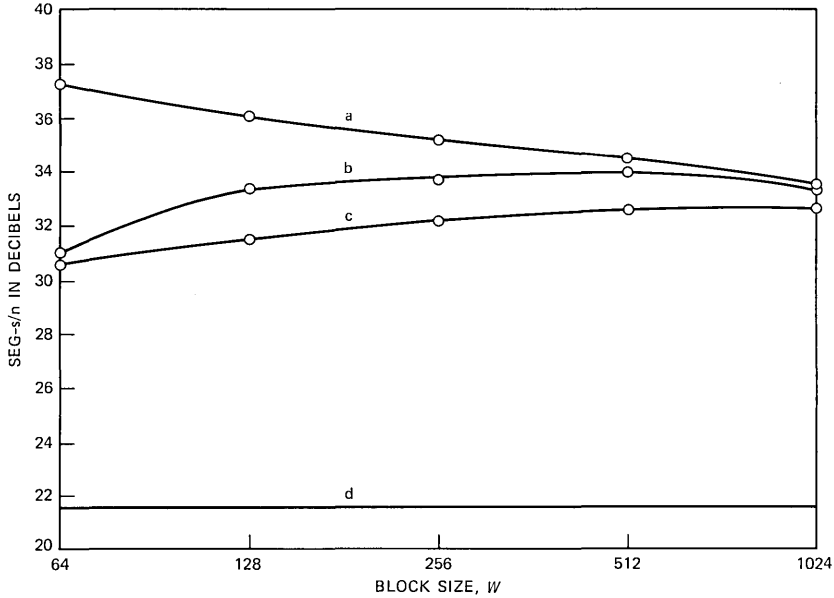


Fig. 6—Variation of SEG-s/n versus block size W for $\lambda = 3$, $n = 4$. The correlation function was computed for curves a, b, and c using eqs. (12), (16), and (18), respectively. Curve d applies to nearest neighbor averaging.

blocks. By contrast the SEG-s/n determined using $R(\tau)$ deteriorates with decreasing W. Now $R(\tau)$ is the conventional correlation function that assumes the speech signal to have stationary statistics. For the larger block sizes shown in Fig. 6, local statistical departures from stationarity tend to be smoothed by the $R(\tau)$ equations, but at low values of W a considerable number of interpolation errors occur in some blocks to yield a low SEG-s/n. When $W = 1024$ the SEG-s/n values computed using the different $R(\theta)$ expressions are very similar. We do not plot curves for $W < 64$ as the side information to transmit the interpolation parameters is unacceptably high [see eq. (20)], and for $W > 1024$ the delay is excessive (>250 ms). Thus, by using $R(k, j)$ we obtain higher and better interpolation performance compared with employing the conventional $R(\tau)$, but the computational complexity is greater.

6.2 Interpolation parameters generated from the received sequence

Having concluded that every fourth speech sample can be discarded and replaced by an interpolated sample to yield speech with negligible perceptual degradation, we next considered the performance of our scheme when the interpolation parameters were derived from the received data. The problem in this case was how to obtain a reliable estimate of the autocorrelation function $R(\tau)$. The procedures described in Section V for determining $R(\tau)$, and thence the interpolation parameters, were tried, and the variation of SEG-s/n with block size

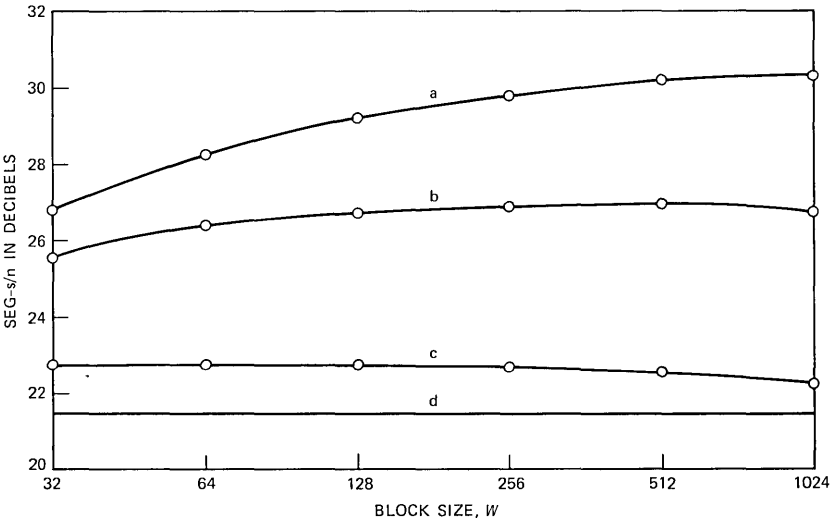


Fig. 7—SEG-s/n versus block size for interpolation parameters derived from the received sequence. Curves a, b, c, and d apply for λ-with-iteration, λ = 3; λ-interpolation, λ = 3; λ-interpolation, λ = 1; and average of adjacent sample interpolation, respectively.

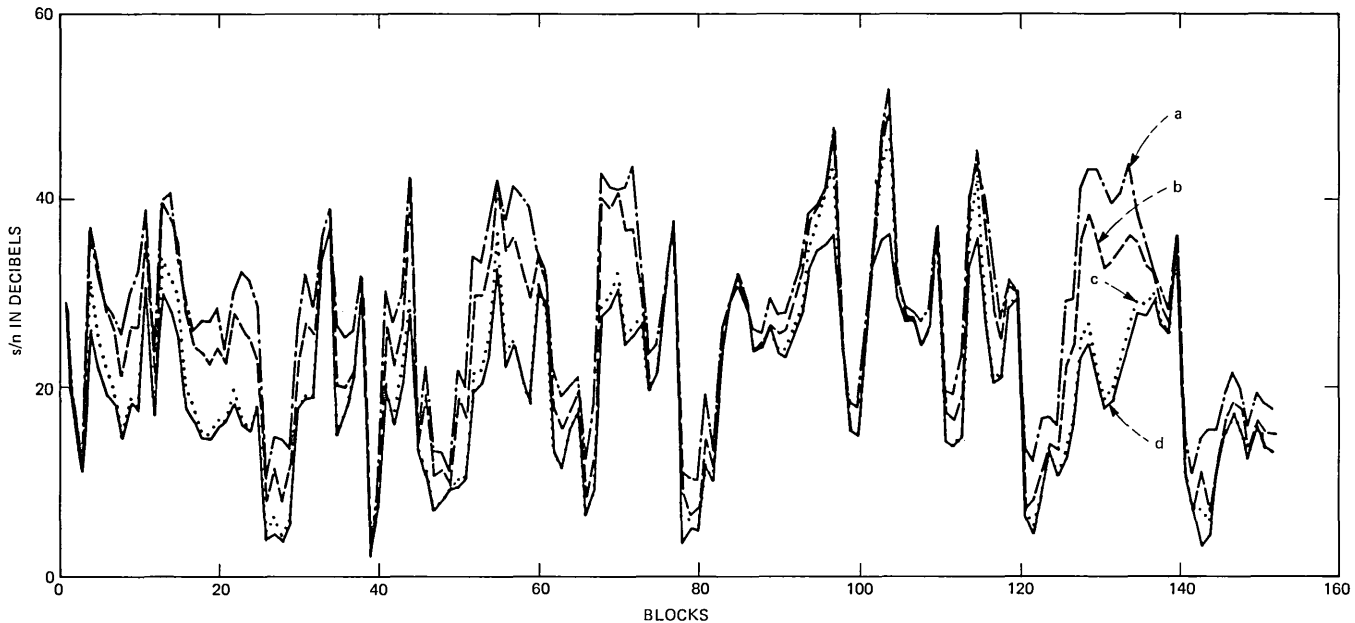


Fig. 8—Variation of block s/n versus block number for the iterative interpolation procedure, $W = 256$, $n = 4$. Curves a, b, c, and d apply for $\lambda = 3$, 2, 1, and nearest neighbor averaging, respectively.

W found for different values of λ. Curve d in Fig. 7 shows that a SEG-s/n of 21.5 dB was obtained when the interpolation process used nearest neighbor averaging, and was employed as a reference level. When the nearest neighbor interpolation was made adaptive, eq. (23) was used for which $\lambda = 1$, and curve c obtained. By using the iteration procedure where the interpolation was made according to eq. (25), $\lambda = 3$, a s/n > 30 dB was achieved for $N = 512$ (see curve a).

Also shown in Fig. 7 is curve b, which was obtained by formulating the sequence $\{\hat{z}_k\}$ based on first computing $\{\bar{z}_k\}$ according to eq. (22), formulating $R(\tau)$, and then removing samples $\bar{z}_r$ and replacing them with interpolated samples derived by using $\lambda = 3$. Thus, in this λ-interpolation method we do not progress from the average sequence to those derived with $\lambda = 1$ and 2, but proceed directly from the average sequence to compute the parameters with $\lambda = 3$. The result is a s/n of 27 dB for $W = 256$ to 1024, a 3-dB reduction compared to the λ-with-iteration ($\lambda = 3$) case, and a diminution in complexity. Informal listening experiences showed that the λ-interpolation and λ-with-iteration schemes, both with $\lambda = 3$, produced speech whose impairments were barely perceptible.

The variation of block s/n with block number for the iterative interpolation procedure is displayed in Fig. 8, $n = 4$, $W = 256$. The average values of these block s/n's give the segmental s/n in Fig. 7 for $W = 256$. Close inspection of the curves in Fig. 8 reveal that progressive iteration does not always give the highest block s/n. However, λ-with-iteration, $\lambda = 3$, achieved the highest s/n for most blocks with gains up to 25 dB compared to nearest neighbor averaging interpolation.

In Fig. 9, the variation of block s/n with block number is displayed for the $\lambda = 3$ condition, $n = 4$, $W = 256$. Curves a and b are those previously displayed in Figs. 5a and 8a, and refer to interpolation parameters computed from the original speech sequence, and by λ-with-iteration procedure, respectively. Curve c applies for the λ-interpolation method, while curve d corresponds to nearest neighbor averaging interpolation, and is included as a reference level. The curves in Fig. 9 illustrate that by deriving the interpolation parameters from the original speech instead of from the received data, the large dips in block s/n are mitigated.

6.2.1 Interpolation errors

We will now consider the interpolation error sequences and their spectra when the interpolation parameters are generated from the received sequence $\{z_k\}$. To illustrate the error performance we selected an arbitrary segment of our input speech signal (see Fig. 3) that had high-level and low-level voiced speech, and unvoiced speech. The speech segment and its spectrogram are displayed in Fig. 10. As before,

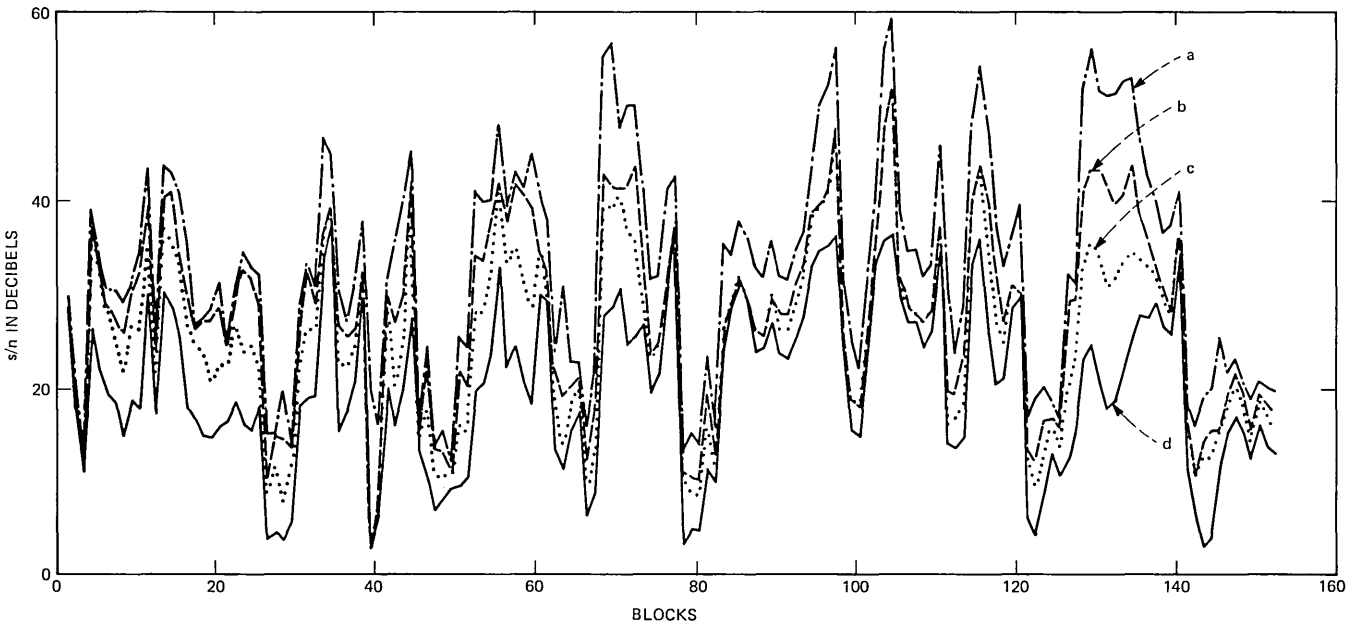


Fig. 9—Variation of block s/n as a function of block number for the $\lambda = 3$ condition, $W = 256$, $n = 4$. Curves a, b, c, and d are for interpolation parameters computed from original speech, λ-with-iteration, λ-interpolation, and nearest neighbor averaging, respectively.

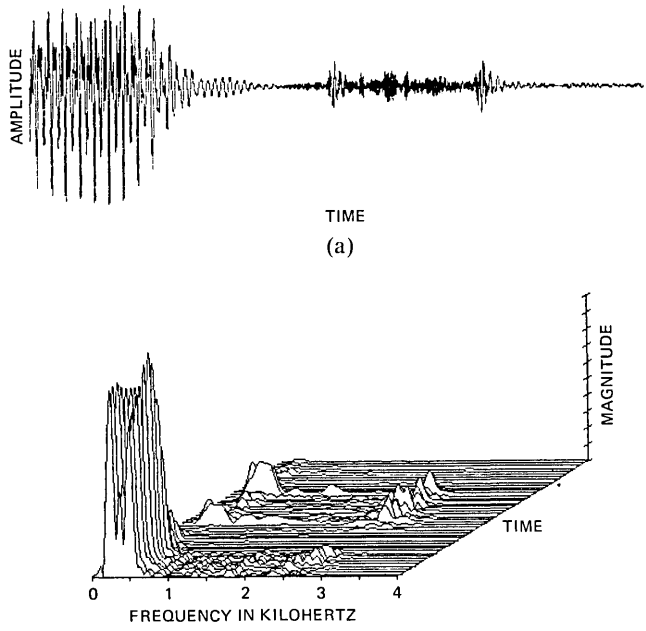


Fig. 10—Speech segment: (a) time waveform, and (b) spectrogram.

every fourth sample in the segment was removed and replaced by a sample produced by interpolation from neighboring samples, where the interpolation parameters were derived from the received data as described in Section V. We avoid displaying the spectrograms of the recovered speech segments associated with the four interpolation conditions used in Fig. 7 because of their similarity. Instead we show in Fig. 11a, b, c, and d the error signals determined as the difference between the input speech segment shown in Fig. 10a and the recovered waveforms produced using interpolation methods of: average of adjacent samples; λ-interpolation with $\lambda = 1$ and 3; λ-with-iteration, $\lambda = 3$; respectively. The error signals in Fig. 11 are small for the low-level highly correlated voiced speech section, and are much greater in the high-level voiced section and for the unvoiced speech. The block s/n values for the speech signal in Fig. 10a are therefore higher for the voiced speech than for the unvoiced speech. Inspection of Fig. 11 shows that the smallest error signal amplitudes generally occurred when the interpolation procedure used three samples on either side of each missing sample. For this segment of speech the advantage of using the full iteration procedure is small compared to $\lambda = 3$, no

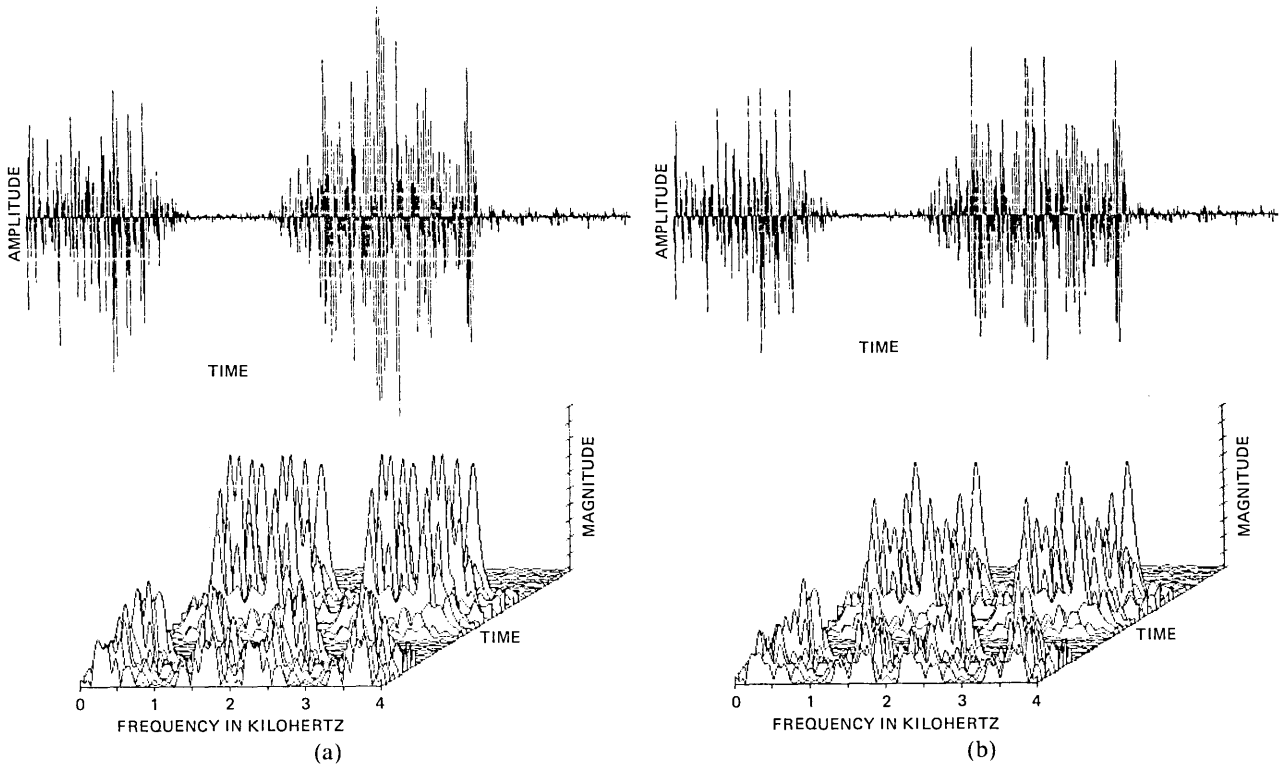
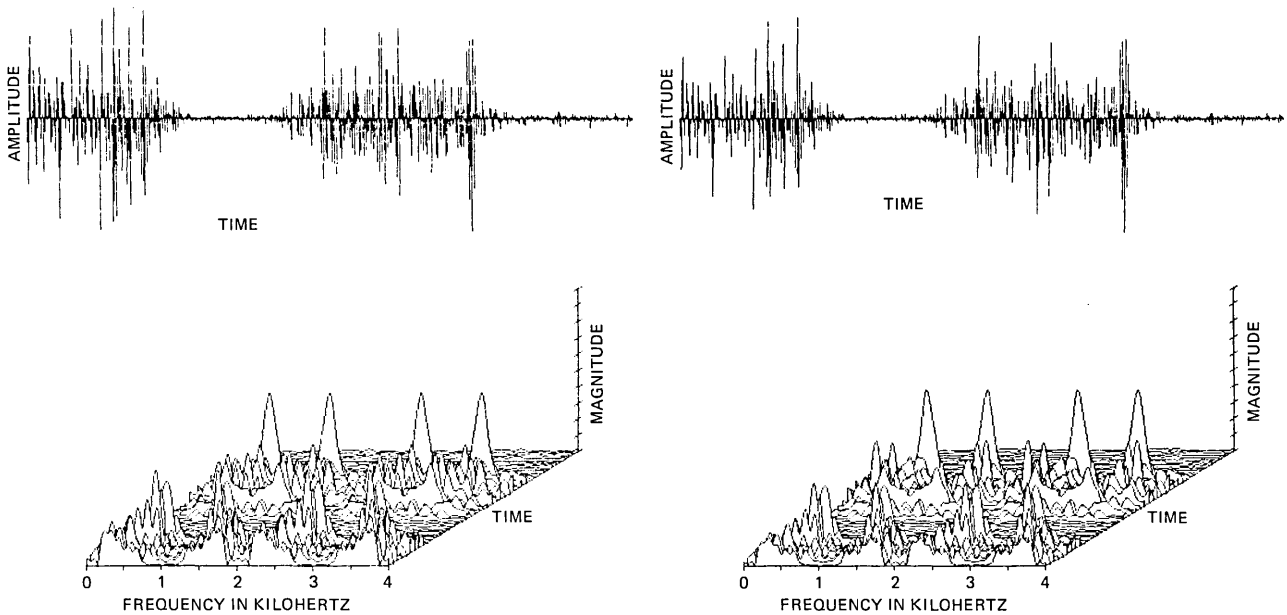


Fig. 11 (a) and (b)—Error waveforms and their spectrograms.

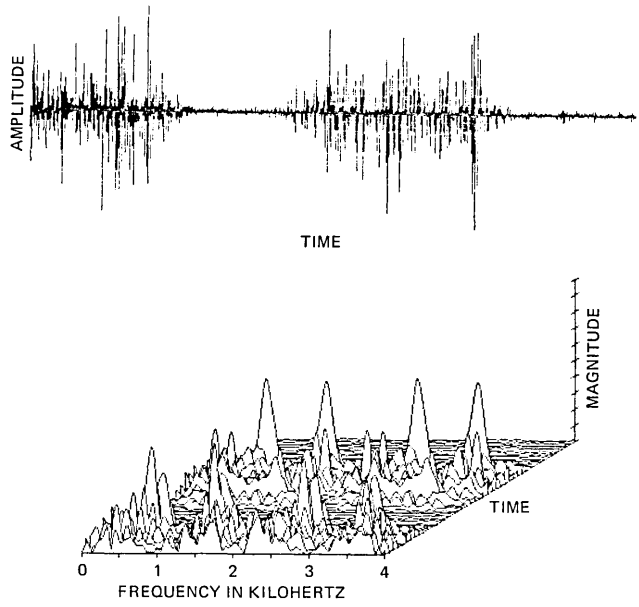
(Figure is continued on next page.)



(c) (d)

Fig. 11 (c) and (d)—Error waveforms and their spectrograms.

(Figure is continued on next page.)



(e)

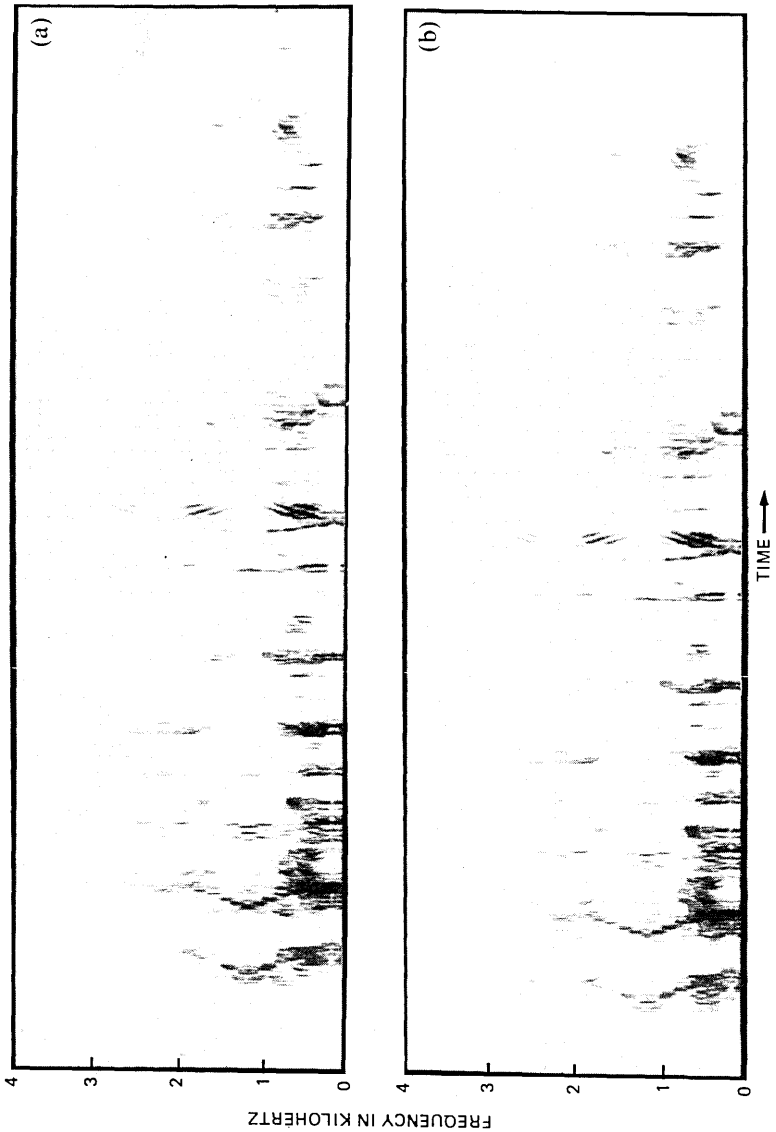
Fig. 11 (e)—Error waveforms and their spectrograms. The error waveforms in a, b, c, and d are the difference between the input speech segment of Fig. 10a and the recovered signals formulated using the interpolation method of: average of adjacent samples, λ-interpolation, $\lambda = 1$, $\lambda = 3$, and λ-with-iteration, $\lambda = 3$, respectively, $W = 256$. The waveform in e shows the effect of μ-law PCM encoding the speech samples prior to interpolation, λ-with-iteration, $\lambda = 3$, $W = 256$.

iteration. The spectrograms of the error signals are displayed above these signals in Fig. 11.

As the error sequence $\{e_k\}$ is the difference between the original speech sequence $\{x_k\}$ and the recovered speech sequence $\{\hat{z}_k\}$, it is composed of $n - 1 = 3$ components of zero magnitude followed by a nonzero component due to the interpolation error. The components in the original and recovered speech sequences occur at a rate $f_s = 8$ kHz, while the components in $\{e_k\}$ are generated at $f_s/n = 2$ kHz in Fig. 11. Close examination of the error spectra in this figure shows a symmetry about 2 kHz over the range dc to 4 kHz, and further symmetries about 1 kHz and 3 kHz for the frequency ranges of dc to 2 kHz, and 2 kHz to 4 kHz, respectively.

The noise components above 3.3 kHz were removed by the output filter shown in Fig. 1. When this was done the improvement in speech quality was minimal, as can be anticipated from the spectrograms of Fig. 11. Also the SEG-s/n values shown in Fig. 7 were only increased by a fraction of a decibel.

Figures 12a through d show the error spectrograms obtained when


(a)
(b)
4
3
2
1
0
TIME
FREQUENCY IN KILOHERTZ

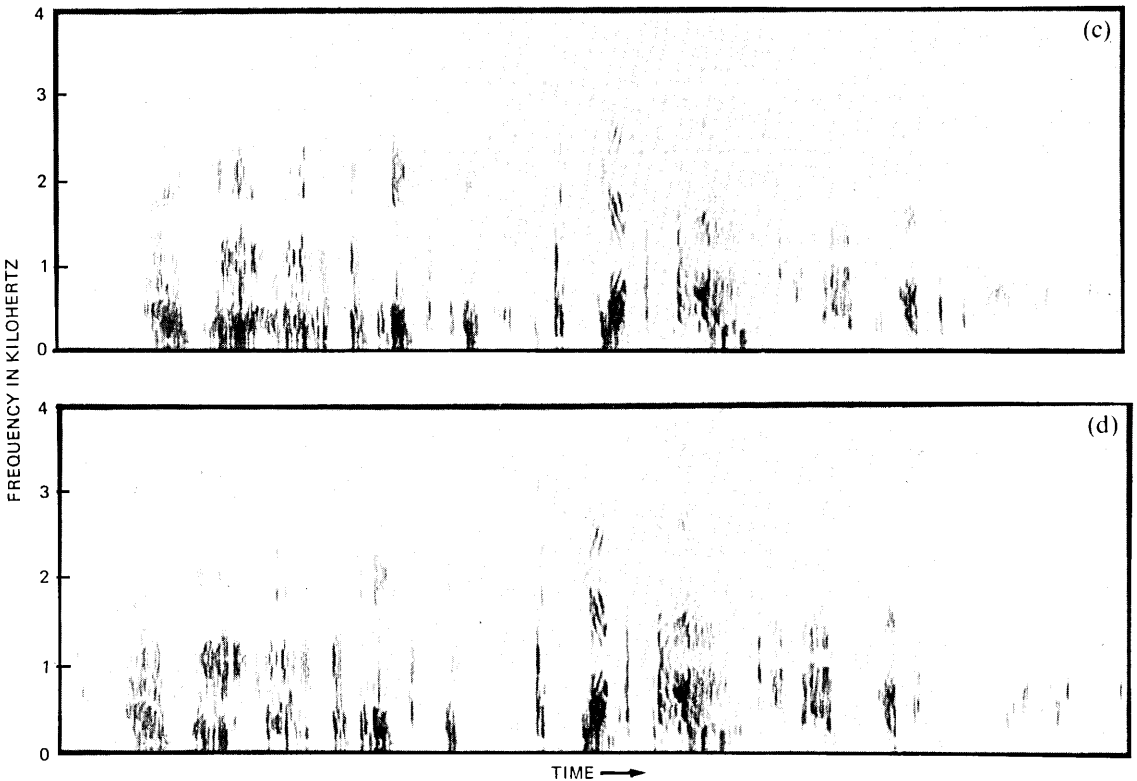


Fig. 12—Error spectograms for the two sentences shown in Fig. 3. The spectrograms a, b, c, and d relate to the conditions a, b, c, and d in Fig. 11, respectively.

the interpolation procedure used adjacent sample averaging, λ-interpolation with $\lambda = 1$, $\lambda = 3$, and λ-with-iteration, $\lambda = 3$ for the entire speech signal of Fig. 3. The spectrograms are not preemphasized. The effect of using $\lambda = 3$ compared to $\lambda = 1$ is to alter the nature of the noise spectrum, increasing its tendency to be more random. When we connected the error signal appertaining to the interpolation procedure having $\lambda = 3$ to a pair of earphones, it was found to be unintelligible and noise-like. With adjacent sample averaging the error signal was significantly more correlated with the original speech, and when we listened to this error signal the two sentences were comprehendible.

6.3 Variation of n

The effect of n on SEG-s/n is displayed in Fig. 13 for a block size of 256. When the interpolation parameters were derived from the original speech sequence using the correlation function given by eq. (14), $\lambda = n - 1$, the SEG-s/n increased by 9 dB to 28.6 dB when n was increased from 2 to 3, as shown in Fig. 13a. Thus, by rejecting every third sample we are able to reconstitute the speech by means of interpolation with only a slight perceptual impairment. For $n = 4$ the distortion in the interpolated speech was imperceptible. Curve b in Fig. 13 applies to λ-with-iteration, $\lambda = n - 1$. When $n = 2$ there is negligible difference between the two curves in Fig. 13, but as n is increased the curves diverge and curve a maintains a minimum mean square interpolation error. We conclude from Fig. 13 that if the block size is 256, the recovered speech will have only minor impairments for $n = 3$ when the interpolation parameters are derived at the transmitter, and $n = 4$ is acceptable when the parameters are computed by the iterative procedure.

6.4 Adaptive interpolation results for PCM encoded speech

The sampled speech sequence $\{x_k\}$ was binary encoded by an 8-bit μ-law PCM encoder having $\mu = 255$. From Figs. 4 through 13 we concluded that $n = 4$ was a good compromise, offering the prospect of an acceptable s/n while reducing the 64-kb/s transmission rate to 48 kb/s. With this bit-rate reduction, 16 kb/s of data can be added to the 48 kb/s of speech to give the conventional transmission rate. At the destination (or at some convenient point along the transmission path) the data can be removed, and each of the 6 k-words/s comprising the μ-law PCM signal decoded. In our experiments we assumed that the bits would be regenerated without error, and the 8-kHz sampling rate established by reinserting the missing samples by the interpolation techniques previously described. However, the accuracy of the interpolation process was reduced by the quantization noise produced in the digital encoder.

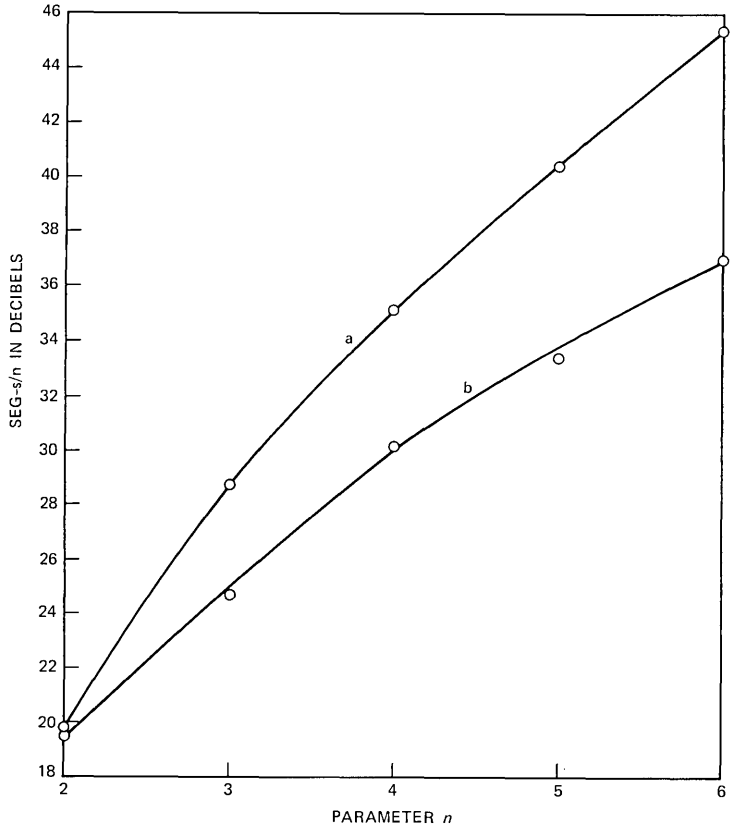


Fig. 13—Variation of SEG-s/n versus parameter n, for $\lambda = n - 1$, and the interpolation parameters derived from the original speech sequence, and by the λ-with-iteration scheme at the receiver. The block size is 256.

Deriving the interpolation parameters from the received data, as described in Section V, yielded the results shown in Fig. 14. The interpolation procedures applicable to curves a, b, c, and d were identical to those employed for curves a, b, c, and d, respectively, in Fig. 7. The respective curves in Fig. 14 are lower than those in Fig. 7 owing to the effect of quantization noise encountered in μ-law PCM encoding. We observe that the quantization noise causes a loss in SEG-s/n of 0.5 dB when the interpolation is performed by adjacent sample averaging. For the cases of $\lambda = 1$ and $\lambda = 3$ the losses in SEG-s/n owing to the effect of quantization noise become approximately 1 and 1.7 dB, respectively, when $W = 256$. As the interpolation process improves, the SEG-s/n becomes relatively more affected by the quantization noise. We observe that the greatest loss in SEG-s/n relative to when there is no quantization noise occurs for the case of λ-with-

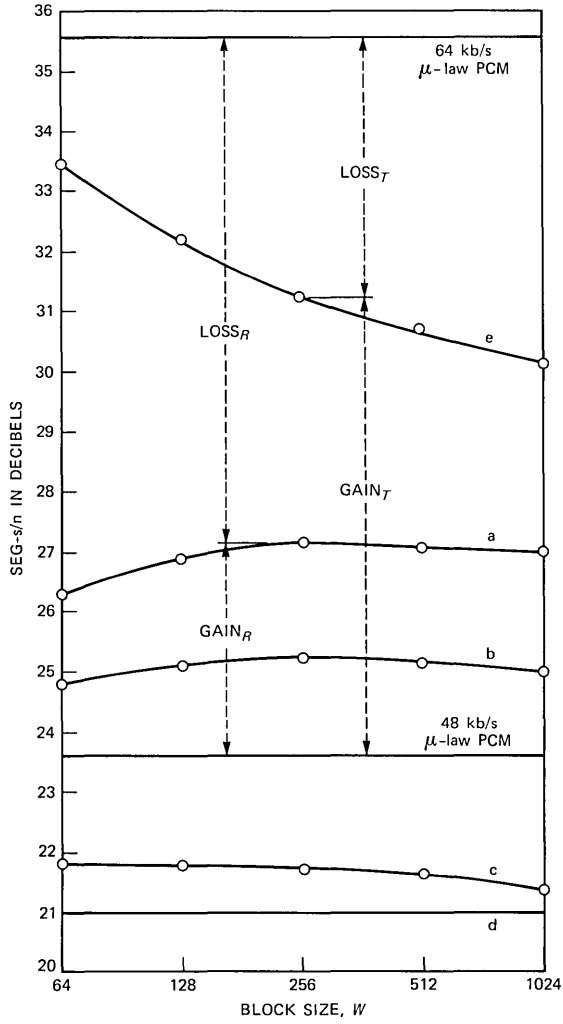


Fig. 14—8-bit μ-law PCM encoding, $\mu = 255$. The effect of block size on SEG-s/n for different conditions. Curves a, b, c, and d apply for λ-with-iteration, $\lambda = 3$; λ-interpolation, $\lambda = 3$; λ-interpolation, $\lambda = 1$; and average of adjacent sample interpolation, respectively. Curve e relates to the interpolation parameters derived at the transmitter using $\lambda = 3$ and $R(k, j)$.

iteration, $\lambda = 3$, and is 2.7 dB for $W = 256$. Also shown in Fig. 14 are the SEG-s/n values for the encoded speech when the sampling rate is 8 kHz, $\mu = 255$, and the number of bits per code word is 8 and 6, i.e., the transmission bit rates are 64 kb/s and 48 kb/s, respectively. The loss in SEG-s/n due to transmitting at only 48 kb/s and reinserting the discarded samples using λ-with-iteration, $\lambda = 3$, compared to not discarding samples and transmitting at 64 kb/s is given by $LOSS_R$ in

Fig. 14. The subscript R indicates that the interpolation parameters were derived from the received data. The gain in SEG-s/n, $GAIN_R$, is due to interpolating the missing samples compared to the 48 kb/s μ-law PCM encoding where no samples are discarded. Although the 48 kb/s has the 8-kHz word rate, the quantization noise is higher because there are 6 bits per code word. The values of $LOSS_R$ and $GAIN_R$ for $W = 256$ are 8.4 and 3.6 dB, respectively.

The interpolation performance can be enhanced if the interpolation parameters are derived from the locally decoded μ-law PCM signal at the transmitter. The interpolation parameters must be conveyed to the receiver in a binary format as side information. This is not a difficult task in a digital transmission system as fewer than 30 bits per W PCM words are required to be transmitted. When W is 256, the side information increases the bit rate by 2 percent. The bit rate can be maintained at 48 kb/s if the side information replaces the least significant bit in every sixth word transmitted. This will have only a marginal effect on the quality of the recovered speech. When the interpolation parameters for $\lambda = 3$ were determined at the transmitter using eqs. (11) through (14) and subsequently transmitted as side information, curve e in Fig. 14 was obtained. Comparing curve a in Fig. 6 with this curve shows that the presence of quantization noise reduces the SEG-s/n of the interpolated speech signal by 4 dB. Nevertheless, curve e is significantly higher than curve a, and the disparity increases to 7 dB for $W = 64$. The $LOSS_T$ and $GAIN_T$ factors, where the subscript T implies the generation of the parameters at the transmitter, had values of 4.3 and 7.7 dB, respectively, for $W = 256$. The effect of interpolation is equivalent to saving more than one bit per word.

For μ-law PCM encoding and the adaptive interpolation procedure of λ-with-iteration, $\lambda = 3$, $W = 256$, resulted in the error waveform and its spectrogram displayed in Fig. 11e for the speech segments shown in Fig. 10. Figures 11e and d show that the effect of quantization on the error spectrum is small.

VII. DISCUSSION

Our intention at the outset of this investigation was to discard one speech sample (or PCM word) in every four, $n = 4$, and to replace the missing samples or words by an interpolation process such that the degradation in speech quality was virtually imperceptible. Further, the implementation algorithm was to be inherently simple. These goals have been reached in good measure.

The central issue in any interpolation process is determining the interpolation parameters. Our approach is to attempt to minimize the mean square error, a nonoptimum procedure for speech signals where

the perception of interpolation noise may be modified by the spectral composition of the speech signal over some 20 ms interval, and temporal effects lasting approximately 200 ms. The justification of the mean square error is based on simplicity, and for $n = 4$ gives good results. In deriving the interpolation parameters of eq. (11) based on $R(k, j)$, we made no assumptions concerning the statistic of the speech signal. The selection of block size W depends upon an acceptable s/n, the need to avoid excessive signal delays resulting from too high a value of W, and the amount of side information permitted, where appropriate, when W is small. Our suggested range of W is from 64 to 1024 (see Figs. 4 and 7). These values of W correspond to durations of 8 to 128 ms, i.e., ranging from approximately a pitch to a syllable period. Computing the interpolation parameters using eqs. (11) and (14), we were able to achieve gains in s/n of 16, 14, and 12 dB compared to interpolation using nearest neighbor linear interpolation for block sizes of 64, 256, and 1024, respectively.

When the estimate of the autocorrelation function could not be based on the original speech sequence, but had to be estimated from the received speech samples where every nth sample was missing, an iterative estimation procedure was employed. By making a crude estimation of the missing samples, the autocorrelation function $R(\tau)$ was computed, and in general a more accurate set of interpolated samples were found. The autocorrelation function was again determined, and the accuracy of the interpolated samples nearly always improved. By this iterative approach, for $n = 4$, λ-with-iteration, $\lambda = 3$ had an interpolation gain over nearest neighbor averaging of 8 dB for $W = 256$, as displayed in Fig. 7.

The effect of discarding every nth sample, $n = 2, 3, 4$, and 5, showed that it is advisable to maintain $n \geq 4$ for imperceptible perceptual degradation. For highly correlated sounds and where some masking of the interpolation noise occurs, we can satisfactorily deploy $n = 3$ and even $n = 2$. However, in general there is considerable distortion power when $n = 2$, which is hardly surprising as half the samples had been rejected. Nevertheless, for $n = 2$ adaptive interpolation yielded a s/n of approximately 19.5 dB, which is noisy but intelligible speech.

Finally, we found that when conventional 8-bit μ-law PCM encoded speech, $\mu = 255$, had its bit rate reduced from 64 kb/s to 48 kb/s to enable 16 kb/s of other data to be transmitted, the recovered speech after decoding and interpolation had a s/n that approximated that of 56-kb/s μ-law PCM.

VIII. ACKNOWLEDGMENTS

The authors thank D. Vitello for her advice with computer simulation, and D. J. Goodman for his constructive criticism of this work.

REFERENCES

1. B. G. Haskell and R. Steele, "Audio and Video Bit-Rate Reduction," Proc. IEEE, *69,* No. 2 (February 1981), pp. 252–62.
2. J. L. Flanagan, M. R. Schroeder, B. S. Atal, R. E. Crochiere, N. S. Jayant, and J. M. Tribolet, "Speech Coding," IEEE Trans. Commun., *COM-27* (April 1979), pp. 710–37.
3. M. V. Mathews, "Extremal Coding for Speech Transmission," IRE Trans. Inform. Theory, *IT-5* (September 1959), pp. 129–36.
4. C. M. Kortman, "Redundancy Reduction—A Practical Method of Data Compression," Proc. IEEE, *55,* No. 3 (March 1967), pp. 253–63.
5. L. Ehrman, "Analysis of Some Redundancy Removal Bandwidth Compression Techniques," Proc. IEEE, *55,* No. 3 (March 1967), pp. 275–87.
6. C. A. Andrews, J. M. Davis, and G. R. Schwarz, "Adaptive Data Compression," Proc. IEEE, *55,* No. 3 (March 1967), pp. 267–77.
7. N. S. Jayant, "Adaptive Aperture Coding for Speech Waveforms-1," B.S.T.J., *58,* No. 7 (September 1979), pp. 1631–45.
8. H. S. Hou and H. C. Andrews, "Cubic Splines for Image Interpolation and Digital Filtering," IEEE Trans. ASSP, *ASSP-26,* No. 6 (December 1978), pp. 508–17.
9. "Special Issue on Bit-Rate Reduction and Speech Interpolation," IEEE Trans. Commun., *COM-30,* No. 4 (April 1982), pp. 728–80.
10. N. S. Jayant, "Effect of Packet Losses in Waveform Coded Speech and Improvements Due to an Odd-Even Sample-Interpolation Procedure," IEEE Trans. Commun., *COM-29,* No. 2 (February 1981), pp. 101–9.
11. R. E. Crochiere and L. R. Rabiner, "Interpolation and Decimation of Digital Signals—A Tutorial Review," Proc. IEEE, *69,* No. 3 (March 1981), pp. 300–31.
12. P. Noll, "Adaptive Quantization in Speech Coding Systems," Int. Zurich Seminar, Zurich, Switzerland, April 1974.

AUTHORS

Frank Benjamin, B.A. (Music Education), 1980, B.S. (Electronic Engineering), 1983, Valedictorian (both cum laude), Monmouth College, West Long Branch, NJ; Monmouth College, 1980-1981; United Telecontrol Electronics, 1981; Bell Laboratories, 1981—. While Mr. Benjamin was attending college, he worked as a recording engineer and studio musician on albums for RCA-Columbia and EPIC studios, and on promotional sound tracks for the motion picture *STAR TREK.*® He taught privately in the fields of music, engineering, mathematics and physics. At Monmouth College he worked as a laboratory instructor and acoustical consultant. While at United Telecontrol Electronics, he helped develop a missile guidance control system. In 1981 he joined Bell Laboratories as a member of the Radio Communications Research Department, designing and writing software simulation systems for speech compression and interpolation techniques. Past President, Lambda Sigma Tau; member, Eta Kappa Nu; nominee, Who's Who Among Students in American Universities and Colleges; New Jersey State Teacher's Certificate.

Raymond Steele, (SM '80), B.S. (Electrical Engineering) from Durham University, Durham, England, in 1959, and the Ph.D. degree in 1975. Prior to his enrollment at Durham University, he was an indentured apprenticed Radio Engineer. After research and development posts at E. K. Cole Ltd., Cossor Radar and Electronics, Ltd., and The Marconi Company, all in Essex, England, he joined the lecturing staff at the Royal Naval College, Greenwich, London, England. Here he lectured in telecommunications to NATO and the External London University degree courses. His next post was as Senior Lecturer in the Electronic and Electrical Engineering Department of Loughborough University, Loughborough, Leics., England, where he directed a research group in digital encoding of speech and television signals. In 1975 his book, *Delta*

Modulation Systems (New York: Halsted), was published. He was a consultant to the Acoustics Research Department at Bell Laboratories in the summers of 1975, 1977, and 1978, and in 1979 he joined the company's Communications Methods Research Department, Crawford Hill Laboratory, Holmdel, NJ. In 1981 Mr. Steele was given *The Bell System Technical Journal* Best Paper Award in the category of Mathematics, Communication Techniques, Computing and Software, and Social Sciences. In 1983 he joined The University of Southampton, Southampton, England, as a Professor of Communications.

THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 6, July–August 1983
Printed in U.S.A.

Practical Design Considerations for Coupled-Single-Amplifier-Biquad Active Bandpass Filters

By J. TOW*

(Manuscript received December 10, 1982)

This paper presents a synthesis method and practical design considerations for the Coupled-Single-Amplifier-Biquad (CSAB) realization of all-pole symmetrical bandpass (BP) filters. The CSAB topology consists of a cascade of second-order SAB bandpass sections, together with negative feedback around adjacent sections. A straightforward procedure that leads to the block diagram representation of the CSAB is shown. Explicit design formulas are given for the optimum element values of the Deliyannis-Friend SAB bandpass section, as well as for the feedback resistors. This CSAB design offers improved performance over the cascade SAB approach without using additional operational amplifiers. Also described are the effects on the filter response due to finite amplifier gain, capacitor dissipations, noninfinite pole-Q sections, and their compensation techniques. These are followed by discussions on maximizing the filter dynamic range and tuning.

I. INTRODUCTION

It is commonly known that in the realization of high-order active filters, properly designed multiple-loop-feedback topologies offer far superior sensitivity performance than the approach of cascading biquadratic filter blocks.[1-6] The multiple-loop-feedback structure is particularly useful in the design of bandpass filters, where reduced sensitivity design is most often needed. Compared with the cascade biquad approach, two drawbacks are usually attributed to these topologies,

* Bell Laboratories, Holmdel, N.J.

namely, the more complicated design procedures and the use of more op amps. However, for the class of all-pole geometrically symmetrical Bandpass (BP) filters, these drawbacks do not exist for a particular multiple-loop-feedback structure commonly referred to as leap-frog, active-ladder, or coupled-biquad.[7-13] The coupled-biquad topology consists of a cascade of second-order sections together with negative feedback around adjacent biquad sections.

Most of the existing coupled-biquad descriptions assumed multiple op-amp biquads, but details on the use of Single-Amplifier Biquads (SABs) are scarce. Our discussion here focuses on the use of SAB and is further restricted to voice-frequency applications. Note that the Coupled-Single-Amplifier-Biquad (CSAB) topology described here requires n op amps for a $2n$-order BP filter.

The first part of this paper presents a general design method for the CSAB realization of all-pole symmetrical BP filters. A straightforward procedure is given that leads to the block diagram representation of the CSAB. Each of the second-order sections is then implemented by the Deliyannis-Friend[14] SAB configuration. Optimum element values for these SABs are computed according to Fleischer's results.[15] The second part of the paper discusses the effects on the filter response due to finite op-amp gain, capacitor dissipations, noninfinite pole-Q sections, and their compensation techniques. These are followed by discussions on maximizing the filter dynamic range and tuning.

II. CSAB DESIGN PROCEDURE

This section presents a straightforward design procedure for the CSAB realization of all-pole symmetrical BP filters. Optimum design equations are given for the Deliyannis-Friend SAB bandpass section and the feedback resistors.

2.1 Block diagram representation of CSAB configuration

The starting point for the CSAB realization of an all-pole symmetrical BP filter, e.g., a Bessel-, Butterworth-, or Chebychev-type BP filter, is its normalized low-pass (LP) prototype ladder configuration. The ladder configuration is readily available from many existing handbooks and is shown in Fig. 1.

Let ω_0 = center frequency of the BP filter (in rad/s)

B = passband bandwidth of the BP filter (in rad/s)

A block diagram representation of the CSAB topology for the BP filter is given in Fig. 2, where the coupled biquadratic transfer functions,

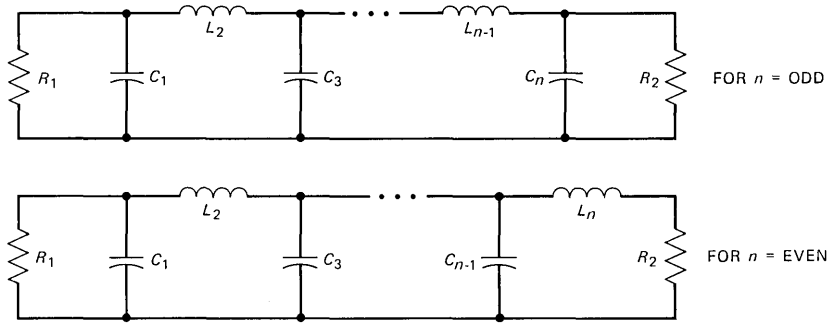


Fig. 1—LP prototype ladder configuration.

$T_i(s)$ and $T_i'(s)$, are related to the element values of the LP ladder by the following equations:

$$T_1(s) = K \cdot \frac{\frac{R_1 + R_2}{R_2} \cdot \frac{B}{R_1 C_1} s}{s^2 + \frac{B}{R_1 C_1} s + \omega_0^2} \tag{1}$$

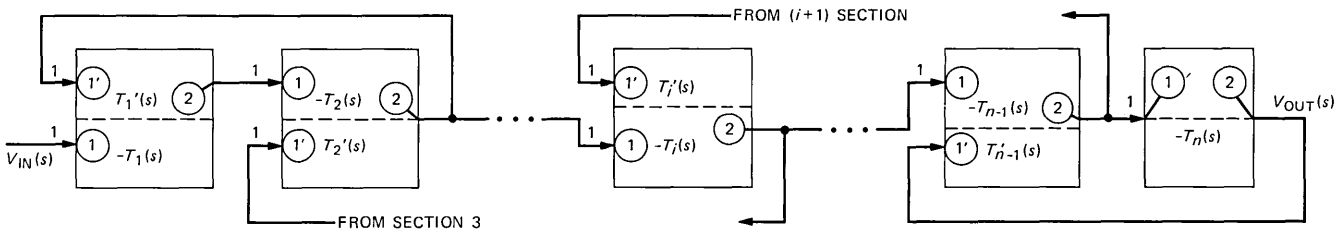
$$T_1'(s) = \frac{\frac{B}{C_1} s}{s^2 + \frac{B}{R_1 C_1} s + \omega_0^2} \tag{2}$$

$$T_i(s) = T_i'(s) = \frac{\frac{B}{X_i} s}{s^2 + \omega_0^2} \qquad i = 2, 3, \cdots, n-1 \tag{3}$$

$$\text{and} \quad X_i = \begin{cases} L_i \text{ for } i \text{ even} \\ C_i \text{ for } i \text{ odd} \end{cases}$$

$$T_n(s) = \begin{cases} \dfrac{\frac{B}{C_n} s}{s^2 + \frac{B}{R_2 C_n} s + \omega_0^2} & \text{for} \quad n \text{ odd} \\ \dfrac{\frac{B}{L_n} s}{s^2 + \frac{B R_2}{L_n} s + \omega_0^2} & \text{for} \quad n \text{ even,} \end{cases} \tag{4}$$

where K determines the overall gain of the filter. For unity (0 dB) voltage gain, set $K = 1$.



WHERE EACH OF THE BLOCKS MAY BE REALIZED BY THE FOLLOWING CONFIGURATION

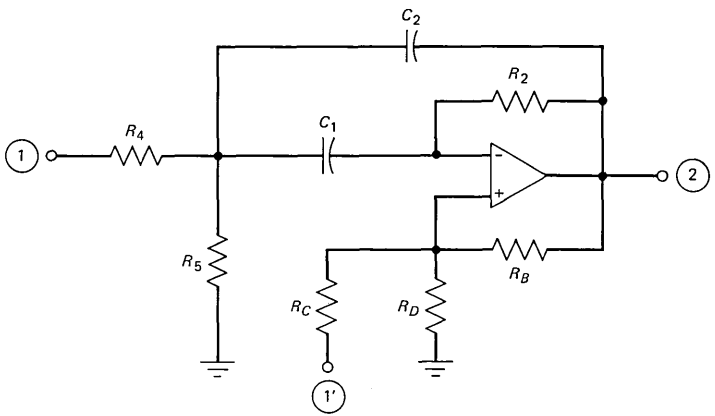


Fig. 2—Coupled SAB configuration for all-pole symmetrical BP filters.

The above derivation is straightforward and is omitted here. For the three-section case, discussions can be found in pages 726 to 729 of Ref. 3 and pages 348 to 351 of Ref. 6. Note that in Fig. 2, the Deliyannis-Friend SAB configuration is also shown. Besides its good sensitivity properties, this particular SAB can simultaneously realize both the forward and the feedback paths as required by Fig. 2.

One final step in the CSAB realization is to obtain the element values for the individual SAB blocks. The optimum formulas are given in the next section.

2.2 Optimum CSAB element values

For each of the second-order blocks shown in Fig. 2, let the forward and feedback voltage transfer functions be represented by:

$$\frac{V_2(s)}{V_1(s)} = -T_i(s) = \frac{-n_1 s}{s^2 + d_1 s + d_0} \tag{5}$$

and

$$\frac{V_2(s)}{V_1'(s)} = T_i'(s) = \frac{n_1' s}{s^2 + d_1 s + d_0}. \tag{6}$$

As shown in Fig. 2 and eqs. (1) to (3), the internal sections, i.e., $i = 2, 3, \cdots, n-1$, have an infinite pole-Q value or $d_1 = 0$. For improved filter performances, a later section suggests the use of a very high-Q value, say several hundreds, instead of the infinite value. This corresponds to the use of a small value for d_1 in eqs. (5) and (6). The high but fixed Q-value can be designed into the CSAB configuration by the familiar predistortion technique.

Element values are first obtained for the forward transfer function, $T_i(s)$, by the formulas given in Ref. 14, where the conductance value of a fictitious resistor, R_1, is first computed and the values of C_1, C_2, R_A, and R_B can be conveniently chosen such that

$$G_1 = \frac{C_2 R_B}{2R_A}\left\{-d_1 + \left[d_1^2 + 4d_0\left(1 + \frac{C_1}{C_2}\right)\frac{R_A}{R_B}\right]^{1/2}\right\} \tag{7}$$

$$R_2 = \frac{G_1}{C_1 C_2 d_0} \tag{8}$$

$$R_4 = \frac{1 + \dfrac{R_A}{R_B}}{n_1 C_2} \tag{9}$$

$$R_5 = \frac{1}{G_1 - G_4} \tag{10}$$

$$R_B = \text{arbitrary [can be chosen according to eq. (15)]} \tag{11}$$

$$R_D = R_A \tag{12}$$

$$R_C = \text{infinite.} \tag{13}$$

As shown by Fleischer,[15] the value of R_B or the ratio of R_A/R_B should be chosen so as to minimize the overall SAB variability (sensitivity) due to the combined active and passive elements variations. His results are given below:*

$$Q_0 = \left[|A(s_0)|^2 \frac{8\sigma_R^2 + \sigma_c^2}{8\sigma_{A(s_0)}^2} \right]^{1/4} \tag{14}$$

$$\frac{R_A}{R_B} = \frac{C_2}{C_1 + C_2} \frac{1}{Q_0} \left(\frac{1}{Q_0} - \frac{d_1}{\sqrt{d_0}} \right), \tag{15}$$

where $A(s)$ represents the gain of the op amp, σ_R^2, σ_C^2, and $\sigma_{A(s)}^2$ correspond to the variances of $\Delta R/R$, $\Delta C/C$, and $\Delta A(s)/A(s)$, respectively, and s_0 is the pole location of the particular SAB transfer function.

The values of R_B and R_C are next modified according to the following formulas to implement the feedback transfer function as well [see Fig. 2 and eq. (6)]:†

$$K_F = \frac{n_1'}{\left(\frac{G_1 + G_2}{C_2} + \frac{G_2}{C_1} \right)} \tag{16}$$

$$\text{(new) } R_C = \frac{1 - K_F}{K_F} R_D \tag{17}$$

$$\text{(new) } R_B = \frac{R_B R_C}{R_C + R_D} \tag{18}$$

$$R_D = R_A \text{ (as before).} \tag{19}$$

In summary, the optimum element values can be computed from eqs. (14), (15), (7) to (12), and (16) to (18).

---

* Equations (57), (61), (13b), (2) and (7) of Ref. 15 were used. For the case of $C1 = C2$ and single-pole op-amp characteristics, simpler formulas are given in page 323 of Ref. 2.

† The derivation is given in the appendix. Note that, as shown in pages 362 to 364 of Ref. 6, the feedback transfer function realized is not exactly a BP function but has negative and real transmission zeros. However, in the vicinity of the pole frequency, where the effect of feedback is of interest, the function behaves like a BP function. In any event, the error introduced is negligible.

III. PRACTICAL DESIGN CONSIDERATIONS

The preceding section describes one practical design consideration, namely, that component statistics are used to obtain an optimum design parameter, R_A/R_B, for the individual SAB blocks. This section further discusses some important factors that cause the SAB or CSAB response to deviate from its ideal characteristic. Compensations in the form of predistortion techniques are described. These compensations can be achieved by modifying the pole locations of the individual second-order blocks as given by eqs. (5) and (6). In a good design, they must be considered as part of the original design and optimum element values are to be computed from the modified transfer functions.

3.1 Nonideal op amp characteristics

For a given SAB realizing the transfer functions (5) and (6), the major effect of finite op-amp gain is to shift the desired pole location, or the roots of

$$s^2 + d_1 s + d_0 = s^2 + \frac{\omega_0}{Q} s + \omega_0^2 = (s - s_0)(s - s_0^*) \tag{20}$$

from s_0 to $s_0 + \Delta s_0$. Fleischer[15] has shown that this deviation is approximately given by

$$\Delta s_0 = -\left(1 + \frac{R_A}{R_B}\right) \frac{1}{A(s_0)} \frac{D_2(s_0)}{s - s_0^*}, \tag{21}$$

where

$$D_2(s) = s^2 + \left(\frac{C_1 + C_2}{C_1 C_2} \frac{1}{R_2} + \frac{G_1}{C_2}\right) s + \frac{G_1}{R_2 C_1 C_2}. \tag{22}$$

In the actual design, one can apply the negative of this shift to the nominal transfer function. Equivalently, as shown in pages 415 to 416 of Ref. 2, the shift in Δs_0 can be represented by:

$$\frac{\Delta\omega_0}{\omega_0} = Re\left(\frac{\Delta s_0}{s_0}\right) \tag{23}$$

$$\frac{\Delta Q}{Q} = -2QIm\left(\frac{\Delta s_0}{s_0}\right). \tag{24}$$

Hence, instead of using the design parameters ω_0 and Q in eq. (20), one can use the modified parameters $\omega_0' = \omega_0 - \Delta\omega_0$ and $Q' = Q - \Delta Q$.

As pointed out by Fleischer, eq. (21) shows that the fractional change in the complex pole location is inversely proportional to the magnitude of the amplifier gain at the pole frequency. Hence, the use of a two-pole, one-zero compensated op amp would provide at least an

order of magnitude smaller pole shift magnitude than that obtained from using a single-pole compensated op amp in most of the audio frequency band.* For many practical voice-frequency-band applications, this pole shift is quite small for the former compensation and its effect can often be ignored.

3.2 Nonideal capacitor characteristics

A nonideal capacitor is usually associated with a finite dissipation factor or tan δ. It is commonly represented by introducing a resistor (with conductance G_i) in parallel with the capacitor C_i, where

$$G_i = \omega_0 C_i \tan \delta_i$$

and ω_0 corresponds to the pole frequency of the SAB block, since we are particularly interested in the variations of the transfer function for frequencies near ω_0.

As with the op-amp finite gain, the major effect of the capacitor dissipation factor is to cause a shift in the desired pole locations, which again can be compensated for. Weyten's approach[16] is described here. For a two-capacitor biquad section realizing eq. (5), Weyten has shown that

$$\frac{\Delta d_1}{d_1} = Q(\tan \delta_1 + \tan \delta_2) \tag{25}$$

$$\frac{\Delta d_0}{d_0} = \tan \delta_1 \tan \delta_2 - \frac{1}{Q}\left(S_{C_1}^{d_1} \tan \delta_2 + S_{C_2}^{d_1} \tan \delta_1\right), \tag{26}$$

where

$$-S_{C_i}^{d_1} = \tfrac{1}{2} + S_{C_i}^{Q} \tag{27}$$

and the coefficients in eq. (20) move from d_i to $d_i + \Delta d_i$. For the SAB under consideration (Fig. 2), Ref. 15 gives

$$S_{C_1}^{Q} = -S_{C_2}^{Q} = \frac{C_2}{C_1 + C_2}\frac{Q}{Q_0} - \frac{1}{2}, \tag{28}$$

where Q_0 is as given before.

Hence, instead of using the design parameters d_1 and d_0 in eqs. (5) and (6), the modified parameters $d_1' = d_1 - \Delta d_1$ and $d_0' = d_0 - \Delta d_0$ can be used. Note that the fractional change of d_0 is usually very small and can practically be ignored. Alternately, eqs. (25) and (26) can be rewritten as

$$\frac{\Delta Q}{Q} \simeq -\frac{\Delta d_1}{d_1} = -Q(\tan \delta_1 + \tan \delta_2) \tag{29}$$

* See for example, pages 84 to 85 of Ref. 6.

$$\frac{\Delta\omega_0}{\omega_0} \simeq \frac{1}{2} \tan \delta_1 \tan \delta_2 - \frac{1}{2Q} (S_{C_1}^{d_1} \tan \delta_2 + S_{C_2}^{d_1} \tan \delta_1) \qquad (30)$$

since the variations of ω_0 are usually very small. The decrease in pole-Q value due to capacitor dissipation factors is usually appreciable and should be compensated for. As an illustration, for a medium-Q BP section, say $Q = 20$, and for a capacitor dissipation factor of 0.0015, such as that encountered with thin-film capacitors, this fractional change in the pole-Q value is 6 percent.

3.3 Infinite pole-Q sections

As we see in Fig. 2, all of the second-order sections except the first and the last have an infinite pole-Q value, i.e., $d_1 = 0$ in eq. (5). Note that the negative feedbacks in the coupled-biquad configuration move these poles away from the $j\omega$-axis and into the desired pole locations. Except for very high-Q BP filters, realization of the infinite pole-Q sections in the CSAB configuration is not very critical as long as the value of these pole-Qs is high, say several hundreds. The effects of using a high but finite pole-Q value in these sections are a lower overall gain of the filter and a reduction of the effective passband bandwidth.[17] A decrease in gain is easily compensated for in active filter design, while the reduced passband bandwidth may usually be absorbed in the original design margin.

On the other hand, a closer approximation to the desired response is obtained if each of the internal sections is a priori designed to have a high but finite pole-Q value. This value can be chosen to be the highest and practically realizable pole-Q value, say Q_M. For the SAB, this is in the order of a few hundreds. Note that in the actual realization, these pole-Q values will deviate a great deal (higher or lower than Q_M). Computer simulations have shown that the overall circuit variability is smaller for the finite internal pole-Q design than the design with infinite pole-Qs.

Having presented the virtues of noninfinite internal pole-Q sections, we show now how this can be achieved with the classical predistortion technique. If we refer back to Section 2.1, instead of starting with the ladder configuration, we see that the overall transfer function for the normalized LP prototype is used (again this is available from many handbooks). A shift α is introduced to the complex frequency variable s, $s = p - \alpha$, where p represents the new complex frequency variable, and

$$\alpha = \frac{\omega_0}{BQ_M}. \qquad (31)$$

In eq. (31) all variables are as defined before. The new transfer function

in p is used to realize the ladder configuration shown in Fig. 1. With the following modifications to eqs. (1) through (4), where R_1' and R_2' are used in place of R_1 and R_2, respectively, the CSAB configuration is again as shown in Fig. 2:

$$R_1' = \frac{1}{\frac{1}{R_1} + \alpha C_1} \tag{32}$$

$$R_2' = \begin{cases} \dfrac{1}{\frac{1}{R_2} + \alpha C_n} & \text{for } n \text{ odd} \\ R_2 + \alpha L_n & \text{for } n \text{ even} \end{cases} \tag{33}$$

$$T_i(s) = T_i'(s) = \frac{\frac{B}{X_i} s}{s^2 + \frac{\omega_0}{Q_M} s + \omega_0^2} \qquad i = 2, 3, \cdots, n-1. \tag{34}$$

In addition, the value of K in eq. (1) must be modified to obtain the desired overall gain of the filter. This value is easily determined by computing the gain of the ladder in Fig. 1 after replacing each inductor L_i with a resistor of value $\alpha^* L_i$ and each capacitor C_i with a resistor of conductance $\alpha^* C_i$.

The ladder realization of the shifted LP prototype transfer function can be obtained by the classical synthesis technique, or with a filter synthesis program.[18] For best sensitivity performance, this ladder must correspond to the maximum power transfer design.[19]

3.4 Maximizing the filter dynamic range

A rule-of-thumb design procedure for maximizing the dynamic range of a high-order filter is to make the maximum voltage output level at the various amplifiers equal. Variations among the various amplifier outputs in the CSAB are generally much smaller than in the corresponding cascade SAB design. For many applications, one may find the CSAB design as obtained before to be satisfactory.

The maximum voltage output levels at the amplifiers usually occur at frequencies within the filter passband or the transition bands. A practical procedure to maximize the filter dynamic range is to evaluate the filter frequency responses (via a computer program) at each of the amplifier outputs and at a set of discrete frequencies chosen from the passband and transition bands. The maximum output values thus obtained are used to rescale the gain of each of the transfer functions given by eqs. (1) through (4). More formally, let

$$G_i = \text{Max}(V_i) - \text{Max}(V_{\text{out}}) \text{ in dB},$$

where V_i (in dB) is the voltage level at the output of the ith amplifier and $\text{Max}(V_i)$ is the maximum value over the chosen set of frequencies.

Compute

$$H_i = 10^{-\frac{G_i}{20}} \qquad i = 1, 2, \cdots, n-1$$

with $H_0 = H_n = 1$.

Multiply each of the forward transfer functions, $T_i(s)$ by K_i, where

$$K_i = H_i/H_{i-1} \qquad i = 1, 2, \cdots, n,$$

and multiply each of the feedback transfer functions, $T_i'(s)$, by K_i', where

$$K_i' = H_i/H_{i+1} \qquad i = 1, 2, \cdots, n-1.$$

3.5 CSAB tuning

When the STAR realization[14] is used to implement the SAB, the manufacturing tuning procedure is to measure values of the two capacitors on the substrate and then compute the resistor values, based on these measurements, from the predistorted transfer function using the equations given in Section 2.2. The resistors are laser-trimmed to these values. The individual SAB blocks are then connected as shown in Fig. 2. In general, this procedure is sufficient for all practical purposes.

For extremely high-precision filter applications where functional tuning may be desirable, the SAB, like any other single-amplifier-biquad configuration, does not exhibit an orthogonal set of tuning parameters. However, when the desired tuning range is small, e.g., during final mop-up trimming, the following tuning sequence is suggested: For the forward BP transfer function, with R_C connected to ground, use R_4 to adjust for the gain at the pole frequency, R_5 for the pole-Q, and then R_2 for the pole frequency. The adjustments for the pole-Q and pole frequency can be monitored by the 45-degree phase-shift points and the 180-degree phase-shift point, respectively. Finally, with R_4 connected to ground, the feedback factor can be adjusted by R_C. As in any coupled-biquad configuration, the CSAB overall response is relatively insensitive to these feedback resistors.

IV. BP FILTERS WITH FINITE TRANSMISSION ZEROS

Extensions of the CSAB design to BP filters with finite transmission zeros, e.g., elliptic-type BP filters, are available. There are two approaches here. The first is to realize these transmission zeros within the individual SAB blocks;[11,12] however, their design procedures are

rather complicated. A second approach is to form these transmission zeros by a weighted sum of the individual SAB BP sections (Fig. 2) with an additional summing amplifier (in a manner analogous to the feedforward technique described in Ref. 9). Simple design formulas exist for this purpose. The second approach is particularly useful and exhibits excellent sensitivity properties for low-order filters, say fewer than five sections.

V. CONCLUSIONS

A straightforward design procedure is given for the coupled-single-amplified-biquad realization of high-order, all-pole, symmetrical BP filters. Practical limitations and their compensation techniques are discussed. Many of these considerations, i.e., optimum choice of element values, nonideal op-amp and capacitor characteristics, are the same for the cascade or coupled designs. With this in mind, the CSAB design procedure is seen to be not more (if not less) complicated than the cascade design, since the individual second-order transfer functions are more readily computed.

The CSAB approach uses the same number of op amps as that of the cascade SAB approach. This number is equal to n for a $2n$-order BP filter and is approximately half the number required by the many inductance simulation techniques, e.g., the two-op-amp Generalized Impedance Converter (GIC) designs. Sensitivity performances of the coupled-biquad may be considered as the best among all the various multiple-loop-feedback topologies[20] and are far superior to the cascade biquad. These observations, together with the fact that the Deliyannis-Friend SAB, from a sensitivity point of view, is as good as any other circuit in the audio frequency band,[15] suggest that the CSAB described here should be the choice for the design of low to medium-high Q (say, $Q < 60$) BP filters in the audio frequency band.*

VI. ACKNOWLEDGMENT

The author would like to thank R. N. Gadenz for the review of this manuscript.

REFERENCES

1. G. S. Moschytz, *Linear Integrated Networks Design*, New York, NY: Van Nostrand Reinhold Company, 1975, pp. 623–53.
2. G. Daryanani, *Principles of Active Network Synthesis and Design*, New York, NY: John Wiley and Sons, 1976, Chapter 11.

* Practical element values constraints and the lack of an orthogonal tuning sequence for the SAB usually limit its maximum realizable stable pole-Q to 30. Note that in the CSAB design, the maximum pole-Q value for the two end sections is usually less than or equal to one-half the maximum pole-Q of the overall design.

3. A. S. Sedra and P. O. Brackett, *Filter Theory and Design: Active and Passive*, Portland, OR: Matrix Publishers, Inc., 1978, Chapters 10–12.
4. L. T. Bruton, *RC-Active Circuits: Theory and Design*, Englewood Cliffs, NJ: Prentice-Hall, 1980, Chapters 9 and 10.
5. L. P. Huelsman and P. E. Allen, *Introduction to the Theory and Design of Active Filters*, New York, NY: McGraw-Hill Book Company, 1980, pp. 292–307.
6. M. S. Ghausi and K. R. Laker, *Modern Filter Design—Active RC and Switched Capacitor*, Englewood Cliffs, NJ: Prentice-Hall, Inc., 1981, Chapter 5.
7. F. E. J. Girling and E. F. Good, "Active Filters," Part 12, Wireless World, *76* (July 1970), pp. 341–5; "The Leapfrog or Active-Ladder Synthesis," Part 13, Wireless World, *76* (September 1970), pp. 445–50; "Applications of the Active-Ladder Synthesis," and "Bandpass Types," Part 14, Wireless World, *76* (October 1970), pp. 505–10.
8. R. L. Adams, "On Reduced Sensitivity Active Filters," in *Proc. 14th Midwest Symp. Circuit Theory*, May 1971, University of Denver, Denver, Colorado, pp. 14.3.1–8.
9. J. Tow and Y. L. Kuo, "Coupled Biquad Active Filters," in Proc. IEEE Int. Symp. Circuit Theory, IEEE Catalog No. 72CHO594-2CT (April 1972), pp. 164-8.
10. L. T. Bruton, "Topological Equivalence of Inductorless Ladder Structures Using Integrators," IEEE Trans. Circuit Theory, *CT-20* (July 1973), pp. 434–7.
11. G. Szentirmai, "Synthesis of Multiple-Feedback Active Filters," B.S.T.J., *52* (April 1973), pp. 527–55.
12. D. Dubois and J. J. Neirynck, "Synthesis of a Leapfrog Configuration Equivalent to an LC-Ladder Filter Between Generalized Terminations," IEEE Trans. Circuit Syst., *CAS-24* (November 1977), pp. 590–7.
13. K. R. Laker, M. S. Ghausi, and J. J. Kelly, "Minimum Sensitivity Active (Leap Frog) and Passive Ladder Bandpass Filters," IEEE Trans. Circuits Syst., *CAS-22* (August 1975), pp. 670–7.
14. J. J. Friend, C. A. Harris, and D. Hilberman, "STAR: An Active Biquadratic Filter Section," IEEE Trans. Circuits Syst., *CAS-22* (February 1975), pp. 115–21.
15. P. E. Fleischer, "Sensitivity Minimization in a Single Amplifier Biquad Circuit," IEEE Trans. Circuits Syst., *CAS-23* (January 1976), pp. 45–55.
16. L. Weyten, "Variation of the Poles of a Second Order RC Active Filter due to a Finite tanδ of the Filter Capacitors," AEU-Arch Elektron Uebertrag, *28* (March 1974), pp. 140–1.
17. V. Belevitch and C. Wellekens, "Internal Equalization in Filters," Circuit Theory and Applications, *1* (1973), pp. 179–86.
18. G. Szentirmai, "FILSYN: A General Purpose Filter Synthesis Program," Proc. IEEE, *65* (October 1977), pp. 1443–58.
19. H. J. Orchard, "Inductorless Filters," Electron. Lett., *2* (June 1966), pp. 224–5.
20. K. R. Laker, R. Schaumann, and M. Ghausi, "Multiple-Loop Feedback Topologies for the Design of Low-Sensitivity Active Filters," IEEE Trans. Circuits Syst., *CAS-26* (January 1979), pp. 1–21.

APPENDIX

Derivation of the Element Values for the Feedback Transfer Function

With ideal op amp, the voltage transfer functions for the SAB circuit of Fig. 2 are given by:

$$\frac{V_2(s)}{V_1(s)} = -\frac{n_1 s}{s^2 + d_1 s + d_0} \tag{35}$$

and

$$\frac{V_2(s)}{V_1'(s)} = \frac{K_F(s^2 + Fs + d_0)}{s^2 + d_1 s + d_0}, \tag{36}$$

where

$$d_1 = \left(\frac{C_1 + C_2}{C_1 C_2}\right) G_2 - \frac{(G_4 + G_5)}{(G_C + G_D)} \frac{G_B}{C_2} \tag{37}$$

$$d_0 = \frac{(G_4 + G_5)G_2}{C_1C_2} \tag{38}$$

$$n_1 = \left(\frac{G_4}{C_2}\right)\left(1 + \frac{G_B}{G_C + G_D}\right) \tag{39}$$

$$K_F = \frac{G_C}{G_C + G_D} \tag{40}$$

$$F = \left(\frac{G_4 + G_5 + G_2}{C_2} + \frac{G_2}{C_1}\right). \tag{41}$$

The element values as given by eqs. (7) through (13) satisfy the forward transfer function, eqs. (35) and (37) through (39). Note that $G_C = 0$.

In the vicinity of the pole frequency, the feedback transfer function, eq. (36), is closely approximated by

$$\frac{V_2(s)}{V_1'(s)} \simeq \frac{n_1's}{s^2 + d_1s + d_0}, \tag{42}$$

where

$$n_1' = K_F \cdot F. \tag{43}$$

To realize the feedback transfer function, the value of G_C must be finite, say G_C'. If we let the value of G_B take on a new value, G_B', and, furthermore, let all the remaining elements take on the values as given before, then the forward and feedback transfer functions, Eqs. (35) and (37) through (43), can be simultaneously satisfied if the following two conditions are met:

$$\frac{G_B}{G_D} = \frac{G_B'}{G_C' + G_D} \tag{44}$$

and

$$K_F = \frac{n_1'}{F} = \frac{G_C'}{G_C' + G_D}. \tag{45}$$

Note that in eq. (45) the coefficient n_1' corresponds to the like coefficient of eq. (6).

The new value of G_C is obtained from eq. (45) and is given by:

$$G_C' = \frac{K_F G_D}{1 - K_F}. \tag{46}$$

Equations (44) and (46) yield the new value of R_B, which is given by:

$$G_B' = \frac{G_B(G_C' + G_D)}{G_D}. \tag{47}$$

Equations (46) and (47) correspond to eqs. (17) and (18), respectively.

AUTHOR

James Tow, B.S. (E.E.), 1960, M.S. (E.E.), 1962, Ph.D. (E.E.), 1966, University of California, Berkeley; Bell Laboratories, 1966—. Mr. Tow has been concerned with computer-aided network analysis and design, and the implementation of practical active filters for telecommunication systems. He is currently engaged in the application of digital signal-processing techniques and the design of digital signal processor integrated circuits. Member, IEEE, Eta Kappa Nu, Phi Beta Kappa.

THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 6, July–August 1983
Printed in U.S.A.

Modal Structure of an MCVD Optical Waveguide Fiber

By A. CARNEVALE* and U. C. PAEK†

(Manuscript received September 1, 1982)

In this paper we compare modal analyses of a Modified Chemical Vapor Deposition (MCVD) optical waveguide fiber and an idealized fiber. The pertinent profile parameters of the idealized fiber are obtained from the MCVD profile. We have validated the concept of an optimum α developed in our previous work. The reduction of the bandwidth of the MCVD fiber is shown to be directly related to the imperfections in the profile inherent in the MCVD manufacturing process.

I. INTRODUCTION

Multimode lightguides, produced by the Modified Chemical Vapor Deposition (MCVD) process, are expected to be of continuing importance to the Bell System. Such lightguides with bandwidths exceeding 1 GHz·km have been produced in the manufacturing plant.

The unique MCVD process produces the index gradient in the core by depositing many layers of material with different compositions, leading to a ripple effect in the profile. Other imperfections in the MCVD process include certain mechanical limitations, difficulties at the core-cladding interface, and the altering of the core center index (burn-off) during the collapse of the preform. These imperfections in the MCVD process lead to a unique modal structure in the fiber, which

* Bell Laboratories, Murray Hill, N.J. † Western Electric Company, Princeton, N.J.

may differ radically from the modal structure obtained from a perfect power law profile. These differences, for a particular fiber preform and idealization, are analyzed and compared. The effects of the imperfections on dispersion and bandwidth are evaluated.

II. EXPERIMENT

Our method[1] for solving Maxwell's equations has been described earlier and we shall not repeat it here. A brief schematic of the process is given in Fig. 1. In the experiment, the perfect power law profile (N_i)

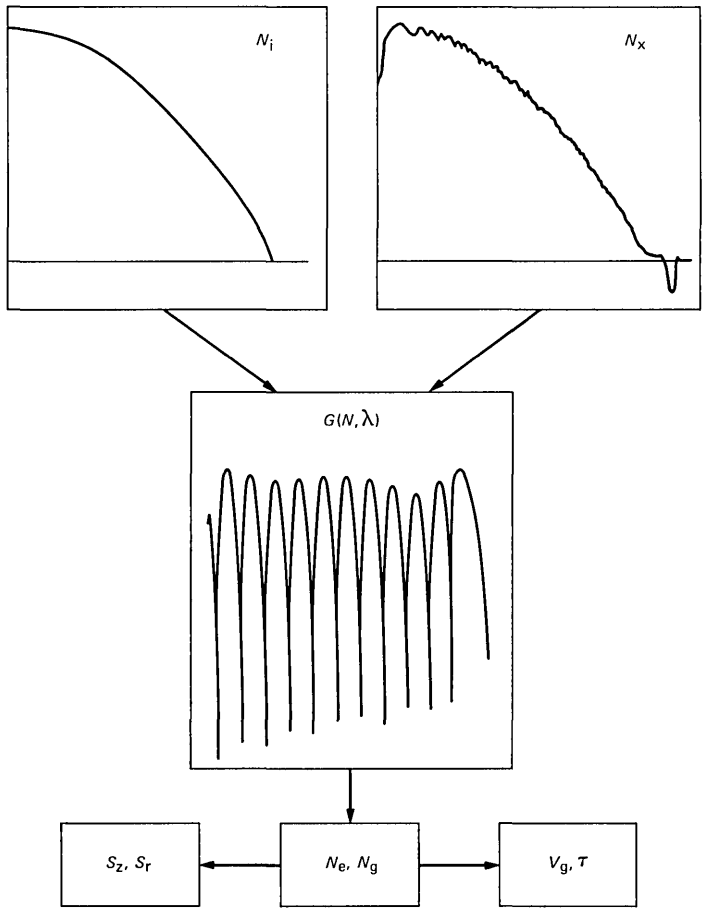


Fig. 1—N_i is a perfect power law profile. N_x is a typical MCVD profile that must be normalized in a particular way to be compatible with the calculations described in Ref. 1. From these calculations we obtain the G function, from which one further determines the effective indices, N_e, and the group indices, N_g. Further processing leads to the calculation of the Poynting vector, the field functions, the delay times, etc.

is replaced with the actual MCVD profile (N_x), which differs from the perfect profile. Thereafter, the data are processed identically to find the effective indices N_e, group indices N_g, the Poynting vector, and other propagation parameters.

An MCVD preform was given to us by S. Jang of the Western Electric Research Center in Princeton, N.J. This preform had been used earlier to produce fiber for experiments. It was typical in the sense that the bandwidths averaged 1 GHz·km. By means of the laser beam refraction method[2] the refractive index profile was determined and is shown in Fig. 2.

We note in this profile the familiar characteristics of the MCVD preform, i.e., a burn-off region in the center, an undefined core-cladding interface region, and ripples on the profile, which are more pronounced as we near the core center.

The pertinent data required for our analytical testing are the approximate $\bar{\alpha}$ and the maximum ΔN. These data can be obtained from known procedures,[2] or alternatively, as in our case, by utilization of the technique described in one of our earlier publications.[3] Briefly, at a selected wavelength, one obtains the modal display for zero azimuthal numbers, fits the data to a linear least squares equation for N_g vs. N_e,

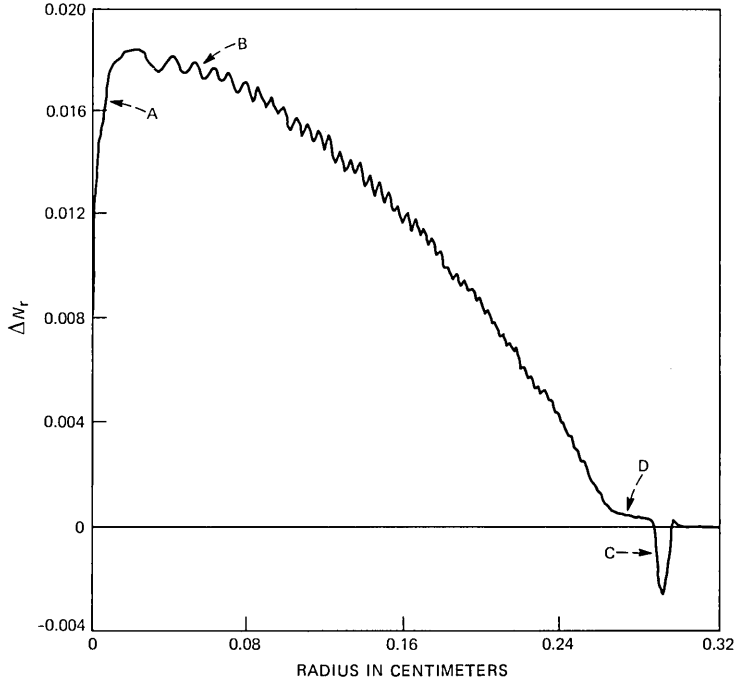


Fig. 2—Typical MCVD profile before normalization.

and compares this slope to the slope vs. α, which has been previously calculated for that wavelength. The parameters to be used in this case were $\alpha = 2$, ΔN is typical for an 11.5-percent GeO_2 dopant, and radius $(R_{cc}) = 25$ μm.

These data are then used to produce modal displays for perfect power law profiles (IDEAL) at the wavelengths and polar indices used for probing the experimental (MCVD) profile. The programs are run in parallel, to be used eventually for comparison purposes only, i.e., there is no communication between the two series of calculations.

Inherent in our calculations is the automatic detection of significant mode splitting between the EH vs. HE, or TE vs. TM, and in the displays to follow, the splittings are shown, wherever possible.

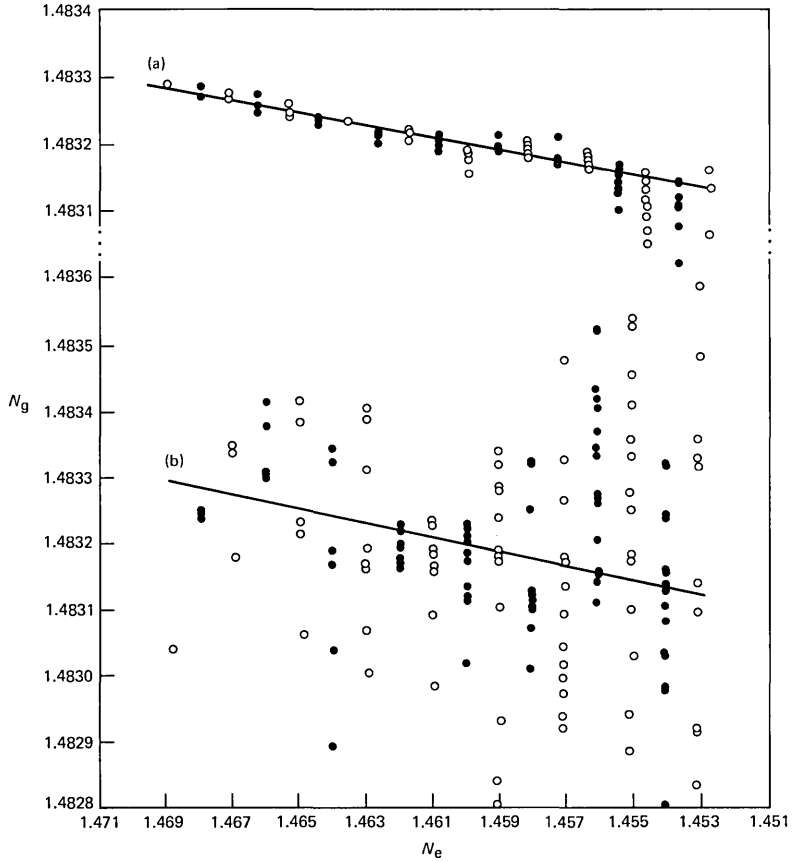


Fig. 3—Composite data for $\lambda = 0.9$ μm. The top portion (a) is the IDEAL. The bottom portion (b) is the MCVD. The straight lines are the linear least-squares curves fitted to all the data in each case, exclusive of those modes near the cladding, and for those modes widely divergent from the line. In general, the open circles are for polar indices $m = 1, 3, 5, \cdots$ last, while the solid circles are for polar indices $m = 0, 2, 4, \cdots$ last. Split modes are not readily identifiable in this display.

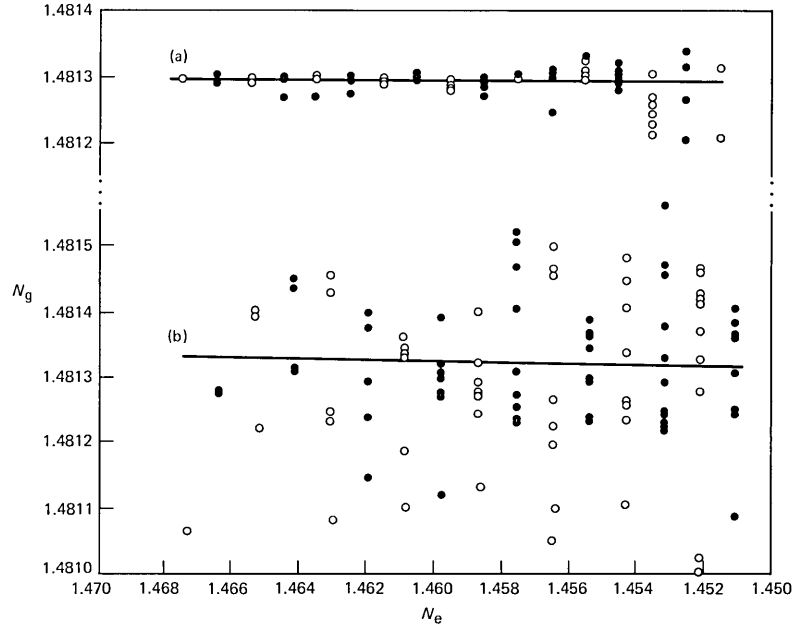


Fig. 4—Same as Fig. 3, $\lambda = 1.0$ μm.

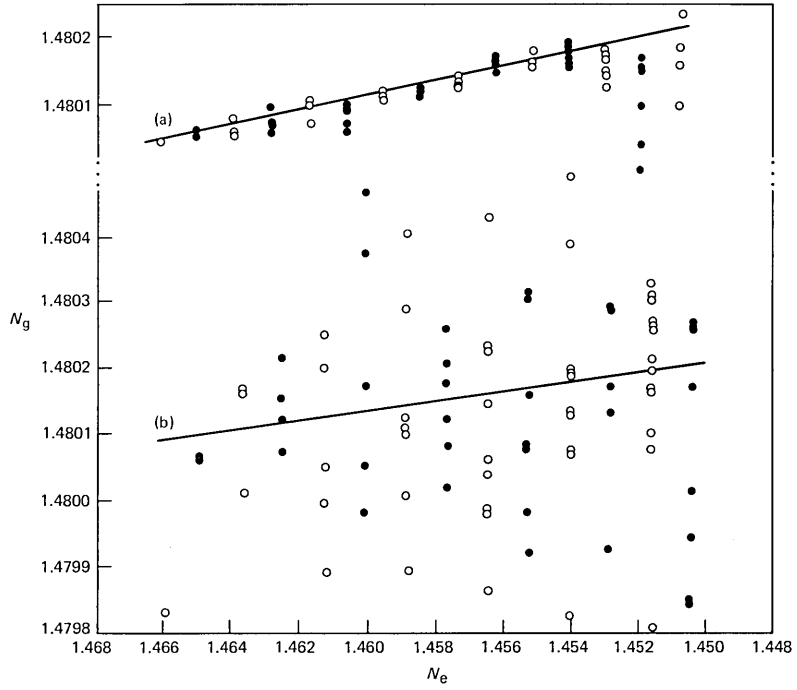


Fig. 5—Same as Fig. 3, $\lambda = 1.1$ μm.

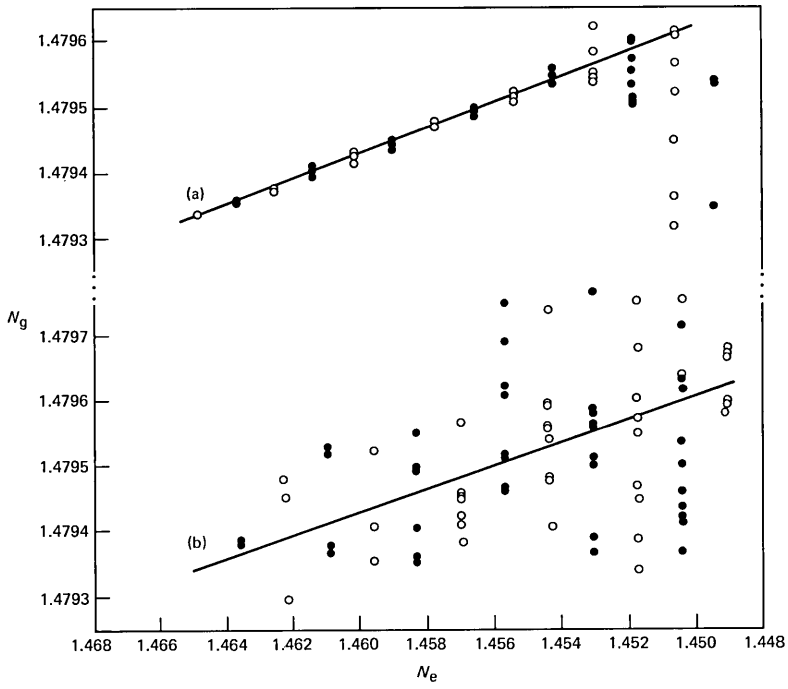


Fig. 6—Same as Fig. 3, $\lambda = 1.2$ μm.

We selected eight wavelengths (λ) spanning the region

$$0.9 \leq \lambda \leq 1.55 \ \mu\text{m}.$$

Each wavelength was checked for all possible modes ($m = 0, 1, 2, \cdots$ last) for both the IDEAL and the MCVD profiles. Because the extremely large amounts of data required in this analysis involved exorbitantly large amounts of computer time, we reduced the required accuracy to a minimum sufficient for our purposes, thereby saving considerable time and expense. Nevertheless, much computational time was required to accumulate the data given in this report.

Our results on the foregoing procedure are displayed in Figs. 3 through 10. In each figure, the dots represent the modes actually determined, the line is the linear least-squares curve fitted to the data, excluding some of the modes near cutoff, the upper portion (a) is for the IDEAL profile, and the lower portion (b) is for the MCVD profile. Again, some modes that are well off the chart are not shown because of space limitations and were not used to calculate the fitted curves in any case. All the fitted curves, translated where necessary but not rotated, have been condensed into the single plot given in Fig. 11.

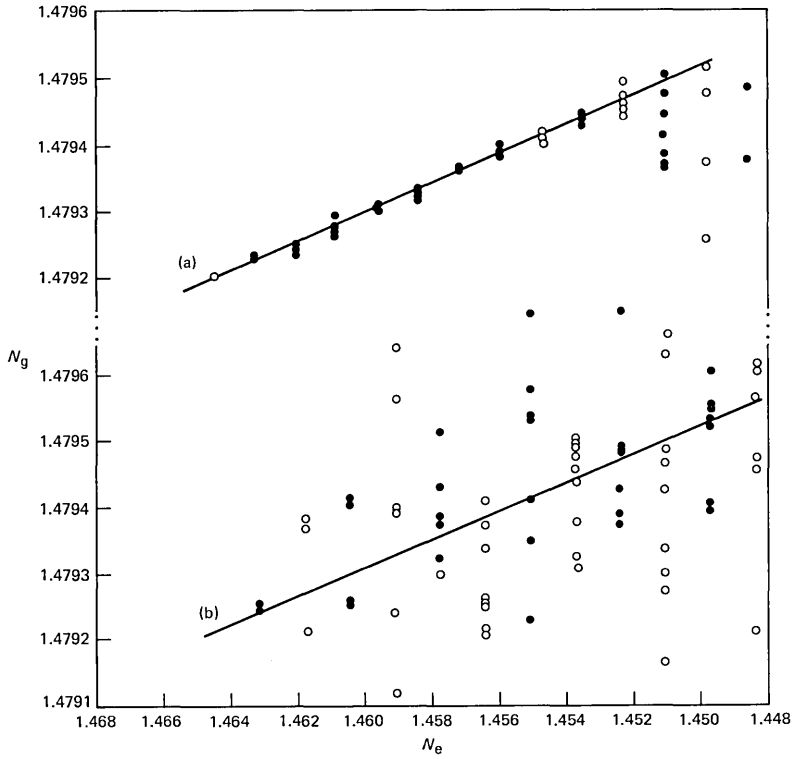


Fig. 7—Same as Fig. 3, $\lambda = 1.23$ μm.

III. DISCUSSION

The degree of scatter in the modes for the IDEAL case (Figs. 3 through 10, Part a) is due in part to the inherent splitting of the HE-EH modes, which has been further compounded by the reduced accuracy used in this analysis.

The scattering of the modes for the MCVD case (Figs. 3 through 10, Part b) is much greater than the IDEAL series, perhaps by a factor of 5 to 10. Obviously, the greatly increased scattering for the MCVD series is due to the imperfections in the MCVD profile, such as burn-off and ripple. The contribution of each of these imperfections to increased modal dispersion (poor bandwidth) has been determined and will be the topic of a forthcoming paper.

Further, within the statistical precision and accuracy employed by the authors, one can calculate that the IDEAL and MCVD data will both extrapolate to the same index at the core. We should realize that the MCVD index of refraction data of Fig. 2 was obtained at $\lambda = 0.6328$ μm. This must be converted to the proper ΔN at the probing

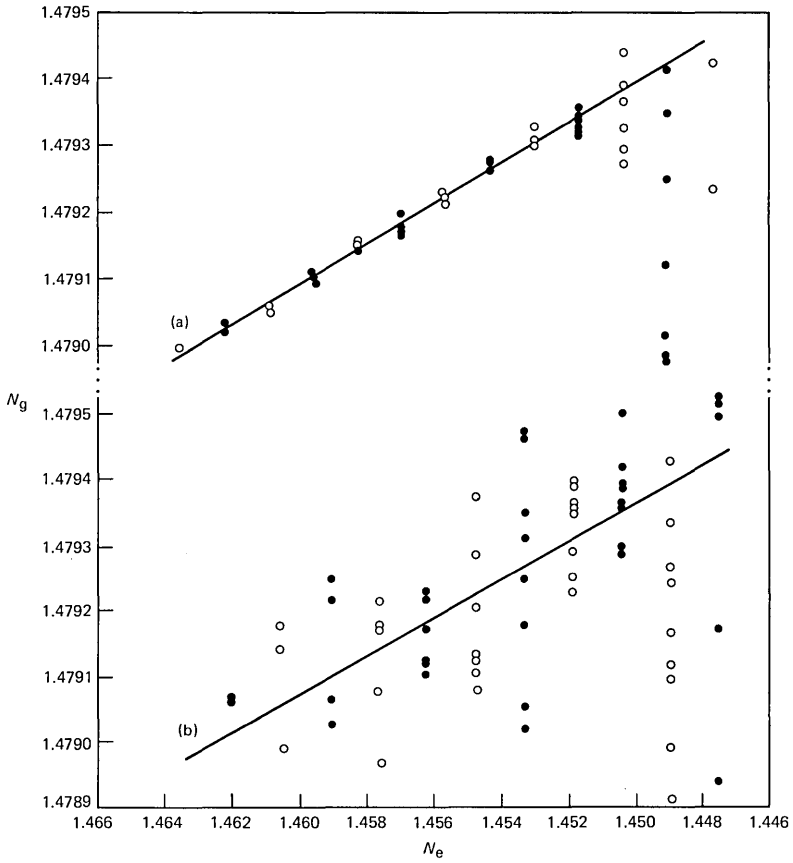


Fig. 8—Same as Fig. 3, $\lambda = 1.32$ μm.

λ's used in any analysis, and was done in this analysis. The results indicate that our parameter ΔN as calculated at each λ is reliable.

The parallelism of the least-squares fitted lines, seen in Fig. 11, indicates that our calculations of the parameter α from the MCVD profile are also reliable, and that an idealized profile may be used to predict characteristics of an MCVD profile. We must realize that α does not vary with λ, and this is true regardless of whether we are using the IDEAL or MCVD profile. We are immediately struck by the rotation of these curves as we change λ. We have noted a similar effect[3] at a fixed λ for a changing α. It should be noted that the previous work[3] and the present work agree closely.

We have one further observation in the present work, which is not readily seen in Figs. 3 through 10: although both the IDEAL and MCVD profiles were normalized to a 25-μm radius, the IDEAL profile

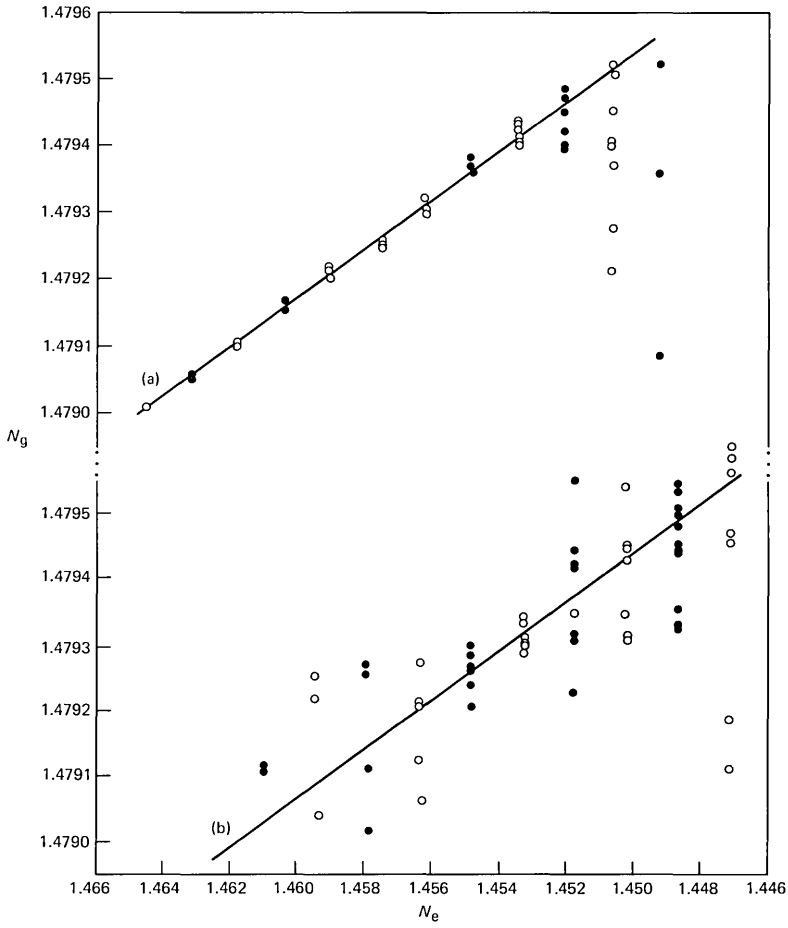


Fig. 9—Same as Fig. 3, $\lambda = 1.40$ μm.

generated more modes than the MCVD, without exception. To account for the fewer MCVD modes a crude estimation of the effective radius for the MCVD indicates that

$$r_e \approx 23 \ \mu\text{m}.$$

It may be only coincidence that this difference of 2 μm is of approximately the same magnitude as the width of the burnout.

IV. REDUCTION OF DATA

Two useful measures of dispersion were derived for this report. The first, which is relatively simple, is the calculation of the dispersion due to the rotation of the line as a function of λ. This implies that every

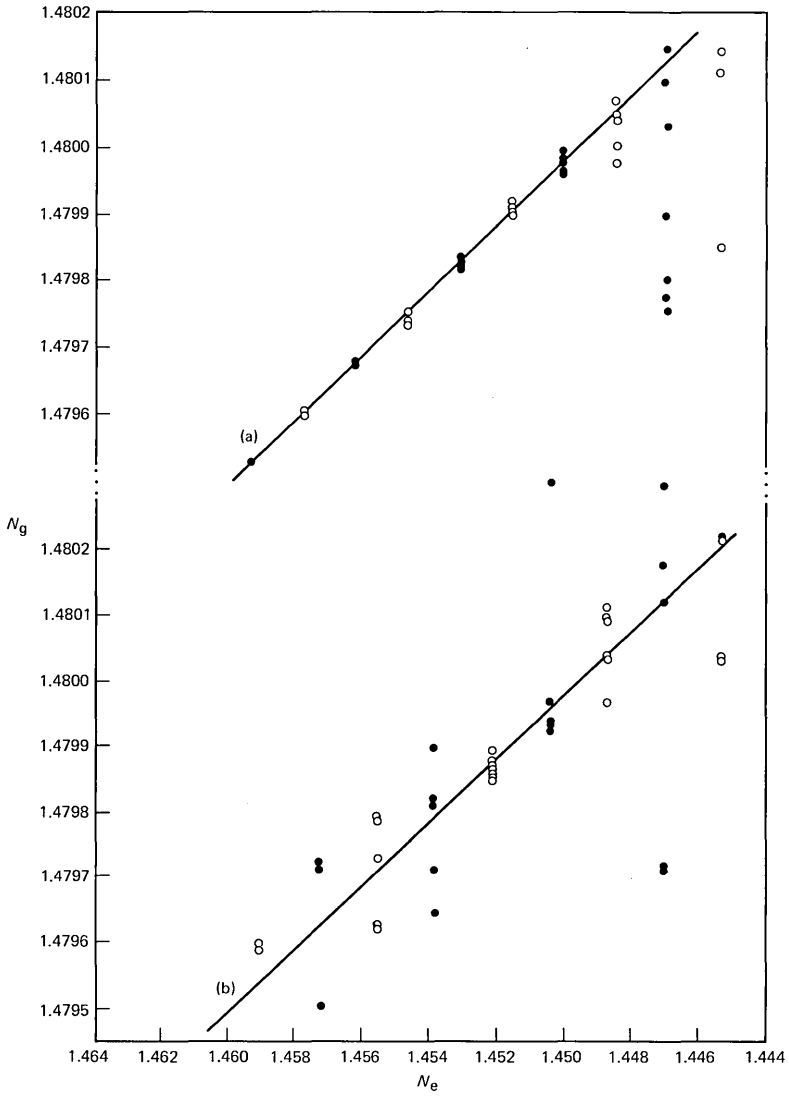


Fig. 10—Same as Fig. 3, $\lambda = 1.55$ μm.

group index (N_g) would be *on* the line. However, the total spread in N_g is now a function of the slope of the line, i.e., the spread in N_g between the $HE_{1,1}$ and the last possible mode that can be obtained on that line. We have given this measure the name "sigma of rotation", (σ_R). Of course, we must count the number of modes at each N_e to find the mean N_g and then calculate σ_R in the conventional manner.

The second measure of dispersion we have utilized is the standard deviation of the group indices determined from the indicated least-

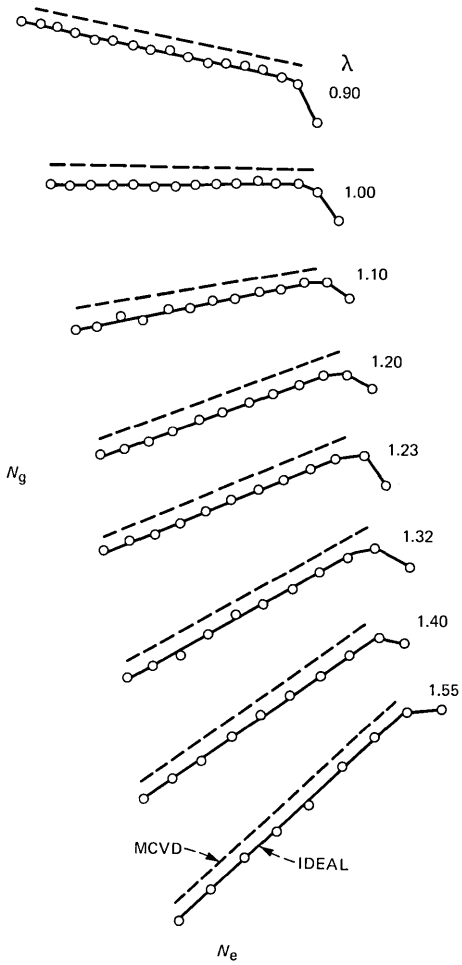


Fig. 11—Composite data. Least-squares fitted lines (IDEAL and MCVD) from the data of Figs. 3 through 10, identifiable in this display.

squares fitted line for N_g vs. N_e. Thus, if the curve value of N_g is denoted by $N_g(\text{curve})$, then

$$\sigma_{N_g} = \sqrt{\frac{\Sigma[N_g - N_g(\text{curve})]^2}{n - 1}},$$

where n is the total number of modes.

The results of these calculations for σ_R are given in Fig. 12, and as indicated in Fig. 11, are about the same for the IDEAL and MCVD fibers. Further, if this were the only measure of dispersion employed,

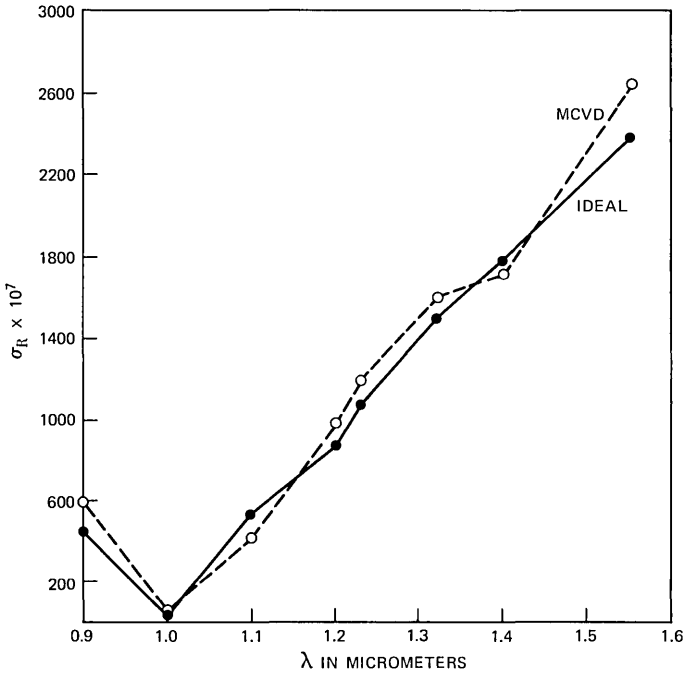


Fig. 12—σ_R vs. λ for IDEAL or MCVD.

we would expect this fiber to yield the maximum bandwidth at $\lambda = 1.00\ \mu m$. This agrees precisely with our previous work.[3] The results for the calculation of σ_{N_g} are given in Fig. 13 and we can see that:

$$\sigma_{N_g}(\text{MCVD}) \approx 10\ \sigma_{N_g}(\text{IDEAL}).$$

Also, we note further that σ_{N_g} is a decreasing function for larger values of λ, and becomes virtually constant when $\lambda \geq 1.2\ \mu m$.

A better understanding of the roles of the two measures of dispersion previously described is given in Figs. 14 and 15. In Fig. 14, σ_R and σ_{N_g} are plotted for the IDEAL case, and in Fig. 15 the same data are plotted for the MCVD case. Apparently, the σ that is dominant at any λ will be the principal cause of inability to maximize the bandwidth.

From theory[4] we have calculated the total dispersion σ_c by:

$$\sigma_c = \sqrt{\sigma_R^2 + \sigma_{N_g}^2},$$

and these data are given in Fig. 16. We see that in the region where σ_R is dominant ($\lambda \geq 1.2\ \mu m$), the curves are roughly parallel. The larger σ_c for the MCVD in this region comes about because the σ_{N_g} of the MCVD is greater than σ_{N_g} of the IDEAL.

Over the range $0.9\ \mu m \leq \lambda \leq 1.2\ \mu m$, the IDEAL case peaks ($\sigma_c =$

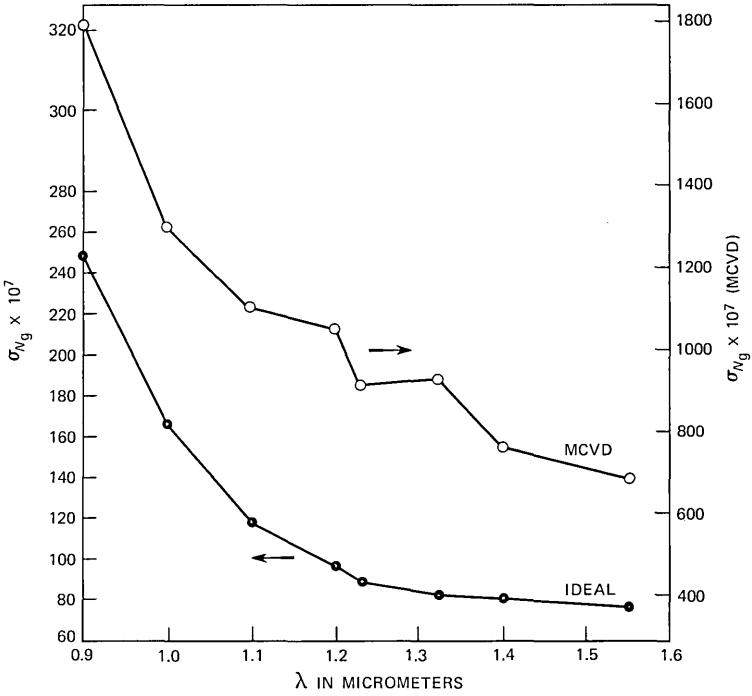


Fig. 13—σ_{N_g} vs. λ for IDEAL and MCVD.

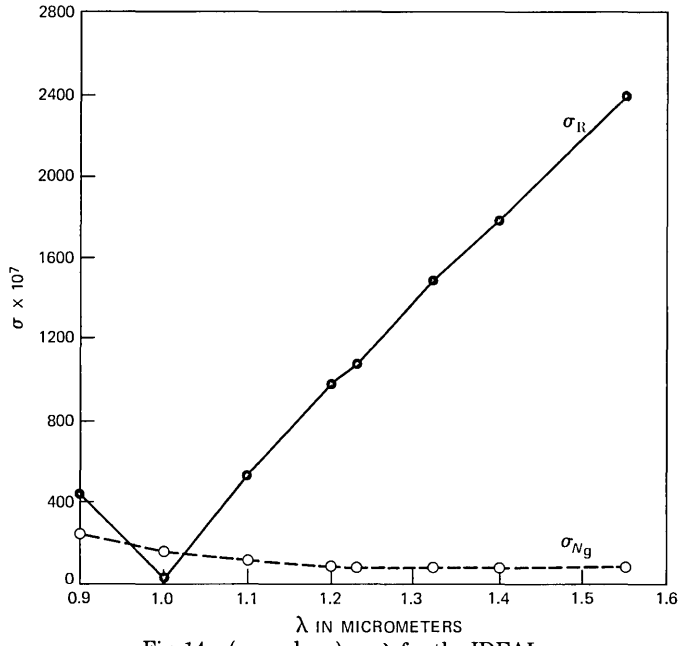


Fig. 14—(σ_R and σ_{N_g}) vs. λ for the IDEAL.

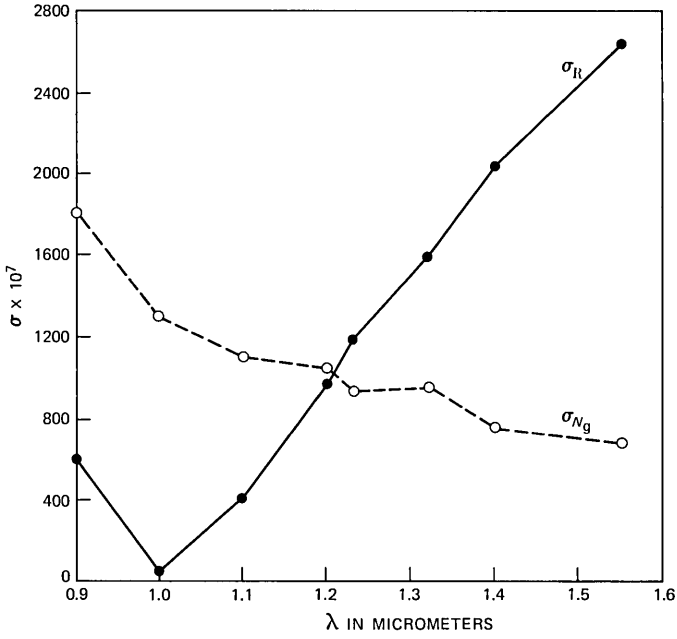


Fig. 15—(σ_R and σ_{N_g}) vs. λ for the MCVD.

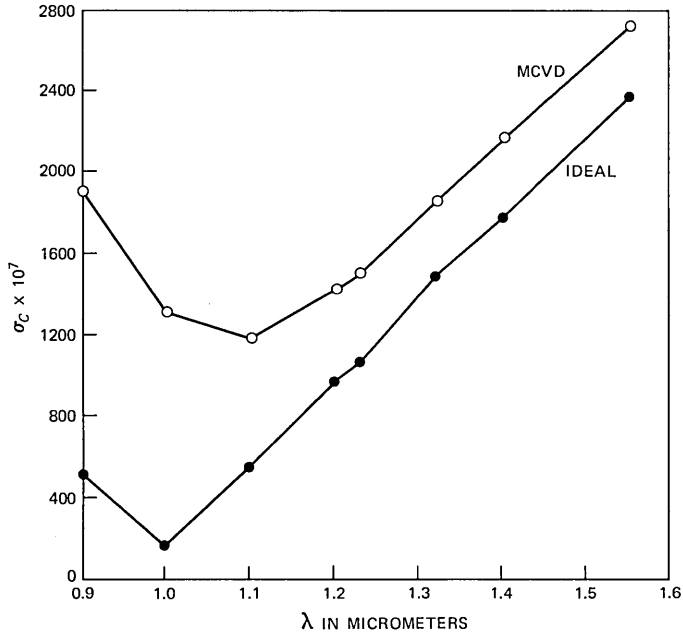


Fig. 16—σ_c vs. λ for IDEAL and MCVD.

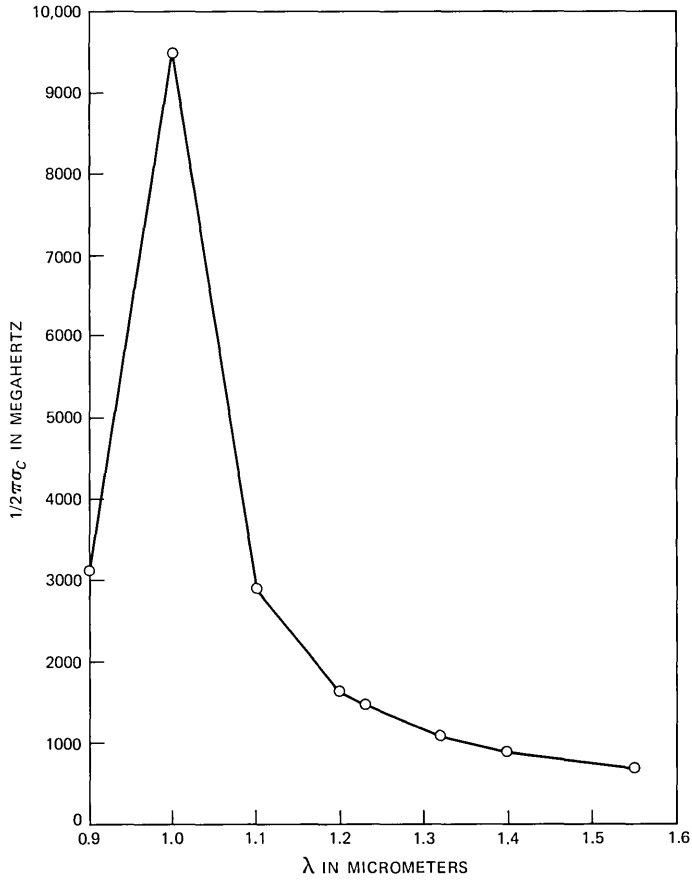


Fig. 17—$1/2\pi\sigma_c$ for the IDEAL case.

minimum) sharply at about 1.0 μm, while the MCVD displays a very broad response that appears to peak at about 1.1 μm. A qualitative measure to estimate the pulse shape can be realized by calculating $W(h/2)$; $W(h/2) = 1/2\pi\sigma_c$ (MHz).

These data are given in Figs. 17 and 18 for the IDEAL and the MCVD, respectively. The IDEAL gives a bandwidth of 9 to 10 GHz/km at 1.0 μm and falls off to half-height at ± 0.7 μm. The MCVD obtains a maximum of about 1.5 GHz·km at ≈ 1.1 μm, and falls off to half height at ± 0.2 μm.

V. CONCLUSION

There can be no doubt that the concept of an optimum α is valid for perfect power law and for experimental (manufactured) profiles.[3] The reduction of the bandwidth is directly related to the imperfections

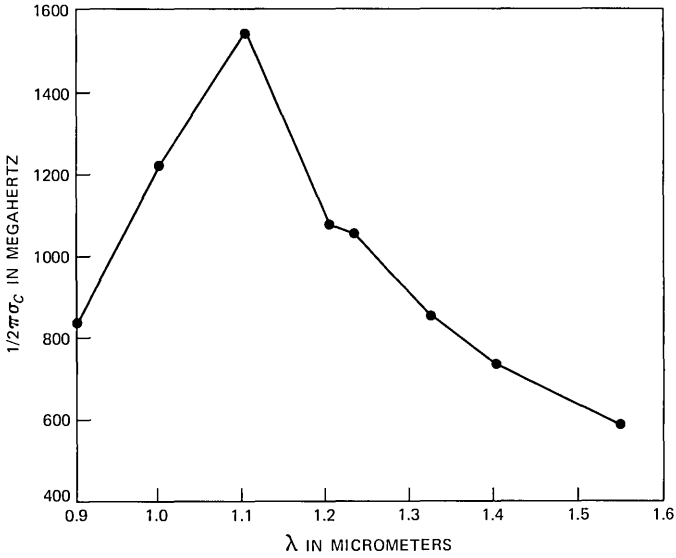


Fig. 18—$1/2\pi\sigma_c$ for the MCVD case.

of the MCVD profile such as burnout, ripples, etc. The direct role of each of these deviations from an idealized profile, in the reduction of bandwidth, has been calculated, and will be the subject of a forthcoming paper. These observations have been substantiated in many respects from direct communication with the materials scientists at Bell Laboratories in Murray Hill and Atlanta, and the Western Electric Company Engineering Research Center. We have also ascertained that we can improve the calculation of σ_{N_g} by about a factor of 2. Thus we find the bandwidth for the idealized fiber is about 18 GHz·km, while for the MCVD fiber it is about 2.8 GHz·km. These bandwidths can be further improved by careful computer-aided design of the profile, for which investigations are currently in progress.

VI. ACKNOWLEDGMENTS

We appreciate the continued support and encouragement given us in this work by M. I. Cohen, R. J. Klaiber, and L. S. Watkins. We are also encouraged by the enthusiasm of the materials scientists engaged in the lightguide program at Bell Laboratories in Murray Hill and Atlanta and Western Electric, as evidenced in a number of informal discussions with them.

REFERENCES

1. G. E. Peterson, A. Carnevale, U. C. Paek, and D. W. Berreman, "An Exact Numerical Solution to Maxwell's Equations for Lightguides," B.S.T.J., *59*, No. 7 (September 1980), pp. 1175–96.

2. L. S. Watkins, "Laser beam refraction traversely through a graded-index preform to determine refractive index ratio and gradient profile," Applied Optics, *18* (July 1, 1979), pp. 2214–22.
3. G. E. Peterson, A. Carnevale, U. C. Paek, and J. W. Fleming, "Numerical Calculation of Optimum α for a Germania-Doped Silica Lightguide," B.S.T.J., *60,* No. 4 (April 1981), pp. 455–70.
4. J. M. Wozencraft and I. M. Jacobs, *Principles of Communication Engineering,* New York, NY: John Wiley & Sons, p. 93.

AUTHORS

Un-Chul Paek, B.S. (Engineering), 1957, Korea Merchant Marine Academy, Korea; M.S., 1965, Ph.D., 1969, University of California, Berkeley; Western Electric, 1969—. At the Western Electric Engineering Research Center, Princeton, N.J., Mr. Paek has been engaged primarily in research on laser material interaction phenomena and fiber optics. Member, Optical Society of America, American Ceramic Society, Sigma Xi.

Anthony Carnevale, B.S. (Physics), 1960, Fairleigh Dickinson University; Bell Laboratories, 1969—. At Bell Laboratories, Mr. Carnevale has been engaged in research on nuclear magnetic resonance, electron paramagnetic resonance, and computer software. For the last four years, his work has been devoted to fiber optics.

THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 6, July–August 1983
Printed in U.S.A.

Asymptotic Analysis of a Queueing Model With Bursty Traffic

By D. Y. BURMAN* and D. R. SMITH*

(Manuscript received September 24, 1982)

Assuming a particular model for "bursty" traffic at a packet-switching node, we find expressions for the expected delay of packets that are valid in light and heavy traffic. Each expression consists of a "correction factor" multiplied by the expected delay experienced by packets when the arrivals are "smooth" (Poisson) and of the same average rate. Approximate values for the correction factor in arbitrary traffic can be obtained by interpolation. This provides an example of a method that often gives fast approximate solutions for bursty traffic models that are not themselves tractable but become so when the offered traffic is assumed to be Poisson.

I. INTRODUCTION

Many models of queueing systems assume that arrivals occur according to a Poisson process. Intuitively, the Poisson process may be characterized by the properties that events occur one at a time and do not depend on the past history of events. Typically, this situation arises when there are large numbers of users of a system, as in the case of arrivals of calls to a central office, since one arrival does not significantly affect the probability of another. Fortunately, these models are often mathematically tractable.

For other systems the Poisson assumptions are not realistic. Often arrivals are indicative of overall activity and give information about the probability of future arrivals. For example, suppose that all arrivals

* Bell Laboratories, Holmdel, N.J.

are generated by a single user who is alternately active and inactive. An arrival indicates that the user is active and hence that there is a greater than average probability of another arrival shortly thereafter. A second example concerns the arrival of packets to a node in a packet-switching network. When virtual calls are employed, the route packets travel for a particular call is fixed for the duration of the call. If we make the assumption that an individual virtual call generates packets according to a Poisson process, then the instantaneous arrival intensity at any node is equal to the sum of the intensities for the calls routed through the node, which varies probabilistically with time. In these examples the arrivals are correlated and the traffic is said to be bursty.

Even the simplest models involving bursty traffic tend to be difficult to solve analytically. Several authors have given approximations. Heffes[1] matched the first three moments of the arrival process to those of an Interrupted Poisson Process. Assuming that an arriving packet requires an exponentially distributed amount of time to be served, he was then able to use the results of Kuczura[2] to analyze this system. Laue,[3] making a similar approximation for the arrival process, assumed that an arriving packet requires a constant service time. The mean waiting time of a packet could then be calculated using the numerical matrix techniques developed by Neuts[4] and Lucantoni and Neuts.[5] A third approach developed by Anick, Mitra, and Sondhi[6] treats a different but related model. The models of Heffes and Laue are based on the assumption that an individual customer generates packets according to a Poisson process. Anick, Mitra, and Sondhi assume that a customer generates packets at a constant rate for a random time. The resulting fluid model was treated numerically by calculating the eigenvalues of the resulting equations.

All of the above works derive numerical techniques to estimate the traffic statistics of interest over a wide range of traffic parameters. It is the goal of this paper to provide simple, closed-form expressions that give insight into the effect of burstiness on delays. This is done by studying queueing systems offered bursty traffic (see Section II for a complete description) in light and heavy traffic.

In light (heavy) traffic, it is obvious that the expected delay tends to zero (infinity). Surprisingly, when the expected delay is divided by the expected delay for the same system offered Poisson traffic of equal average intensity, the ratio goes to a nonzero finite limit in both light and heavy traffic. These limits may be thought of as "correction factors" by which the expected delay for the solvable Poisson system should be multiplied to obtain the expected delay for the bursty system. By interpolating between the light- and heavy-traffic results, one can obtain insight into the approximate effect of burstiness for all values

of traffic. Indeed, similar light- and heavy-traffic limits for the $M/E_k/c$ system have been used to obtain very accurate approximate values of the delay for all values of traffic (see Lauber and Smith[7]).

The remainder of this paper is organized as follows: The queueing model for bursty traffic and our results are discussed in detail in Section II. Rigorous proofs of the light-traffic results are given in Section III when the service times are of the phase type. (Appendix A presents a brief background on phase-type distributions.) Since any service-time distribution can be approximated arbitrarily closely by one of phase type, it is sufficient for practical purposes to establish the light-traffic results for the latter. Appendix B presents an intuitive approach for deriving the light-traffic results. Heavy-traffic results are presented in Section IV and concluding remarks in Section V.

II. QUEUEING MODEL AND RESULTS

The specific queueing model treated here is one in which the arrival process is a nonhomogeneous Poisson process whose rate equals λm, where m is the state of an $M/M/\infty$ queue with birth rate α and service rate β. The arrivals are offered to a single server whose successive services are assumed to be independent and identically distributed according to some general distribution with mean equal to μ^{-1}. Blocked arrivals queue up and are served on a first-in-first-out basis.

This queueing model supports either of two (essentially identical) scenarios for the queueing of packets of information at a packet switch. In both scenarios, the switch is modeled as a single server with an infinite buffer for queued packets, and the service time of a packet is its length (in bits) divided by the line speed. In the first scenario, an individual virtual call generates packets according to a Poisson process with rate λ, for an exponentially distributed length of time with mean β^{-1}. The distribution of requests for virtual calls is Poisson with rate α, and the number of simultaneous virtual calls that can be supported by the switch is unlimited. The second scenario is similar to a fluid model treated by Mitra and Anick[8] and Kosten.[9] In this case an individual customer is in one of two states, either "active" or "inactive." It is assumed that the time in each of the states is exponentially distributed with rates β and γ, respectively. While in the "active" state, the customer transmits packets according to a Poisson process with rate λ. If there are N (large) such customers with $N\gamma \sim \alpha$ and $\gamma\beta^{-1} \sim 0$, then the number of "active" customers is distributed like the number of customers in the $M/M/\infty$ queue described above.

Throughout the remainder of this paper, we will refer to the entities queued at the single server as packets and the entities in the $M/M/\infty$ system as virtual calls or calls. Our analysis will focus on the limiting form for the mean delay of a packet in a lightly or heavily loaded

system. The total number of packets generated during a virtual call is geometrically distributed with mean $\lambda\beta^{-1}$ and the mean total rate of packet generation is $\lambda(\alpha/\beta)$.

There is a technical problem in the analysis for this model since the packet arrival rate is unbounded. Hence, consider first the system where the number of virtual calls is limited to N, i.e., the rate of arrivals of packets is λ times the number of calls present in an M/M/N/N queueing system. The average rate of arrival of packets to the queue is

$$\lambda_N \overset{\text{def}}{=} \lambda(\alpha/\beta)[1 - B(N, \alpha/\beta)], \tag{1}$$

where $B(N, \alpha/\beta)$ is the Erlang Blocking formula. Let $D_B^{(N)}$ be the expected delay of a packet in this system, and let $D_M^{(N)}$ be the delay in an M/G/1 with arrival rate λ_N and the same service-time distribution.

Our key light-traffic result (valid for phase-type service distributions) is

$$\lim_{\lambda \to 0} \frac{D_B^{(N)}}{D_M^{(N)}} = k_l^{(N)}, \tag{2}$$

where $0 < k_l^{(N)} < \infty$, and

$$\lim_{N \to \infty} k_l^{(N)} = 1 + \frac{2\mu}{\alpha} \frac{[1 - \phi(\beta)]}{1 + C^2} \overset{\text{def}}{=} k_l, \tag{3}$$

where $\mu^{-1} = E(S)$, $C^2 = \text{var}(S)/E(S)^2$ (S is a service-time random variable), and $\phi(\cdot)$ is the Laplace transform of the stationary excess of S. The limit (2) is proved by using a simple extension of a lemma established in Burman and Smith.[10] The exact value of $k_l^{(N)}$ is difficult to compute; however, the limit k_l as $N \to \infty$ is computable and can be interpreted as the light-traffic limit of the ratio of the delay (D_B) for the bursty system described earlier (with no limit on the number of virtual calls) and the delay (D_M) for the M/G/1 queue with arrival rate $\lambda(\alpha/\beta)$. This statement can be summarized (although not explicitly proved) as

$$\lim_{\lambda \to 0} \frac{D_B}{D_M} = k_l, \tag{4}$$

where k_l is given in (3).

In the heavy-traffic case no rigorous limit is available. Nevertheless, diffusion analysis gives the following approximation in heavy traffic:

$$\frac{D_B^{(N)}}{D_M^{(N)}} \approx k_h^{(N)}, \tag{5}$$

where $k_h^{(N)}$ is known explicitly [see eq. (40)] and

$$\lim_{N\to\infty} k_h^{(N)} = 1 + \frac{2\mu}{\alpha}\frac{1}{1+C^2} \stackrel{\text{def}}{=} k_h. \tag{6}$$

This supports the conjecture that

$$\lim_{\lambda\to\frac{\beta\mu}{\alpha}} \frac{D_B}{D_M} = k_h. \tag{7}$$

It should be noted that $1 < k_l < k_h$, and while no proof is available, it is reasonable to conjecture that $k_l < D_B/D_M < k_h$ for all stable values of λ. (Indeed it is conjectured that D_B/D_M is monotone, as suggested by a similar analysis for the delay in an M/G/c queue normalized by the delay in an M/M/c queue (see Lauber and Smith[7]).

At this point it is worth noting that the manner in which the traffic intensity $\rho = (\lambda/\mu)(\alpha/\beta)$ approached 0 or 1 affects the limiting value of D_B/D_M. The previously described results are based on the variation of λ only; one can also allow α, the rate of arrival of calls, to vary. For an intuitive example of the difference, note that $\lambda \to 0$ corresponds to light traffic with one packet per call, while $\alpha \to 0$ corresponds to light traffic with a geometrically distributed (with mean λ/β) number of packets per call. It can be shown (although it is not explicitly reported here) that D_B/D_M is completely different in the two cases.

We now focus our attention on the behavior of k_l and k_h for a fixed mean number of calls in the system (α/β). As $\alpha, \beta \to \infty$, (α/β fixed) the arrival process of packets approaches a Poisson process and indeed, by examination of (3) and (6), k_l and k_h both go to 1. As $\alpha, \beta \to 0$ (α/β fixed), k_h goes to ∞, and k_l goes $1 + \beta/\alpha$. To understand these limits, note that in this case the number of calls remains constant for longer and longer periods of time and steady-state effects become significant. In heavy traffic, the process remains in states for which the packet generation rate is faster than the service rate, so that the delays become large. In light traffic, one may obtain the $1 + \beta/\alpha$ limit in an intuitive fashion by conditioning on the number of calls m at packet generation times and computing the conditional delay (using known expressions for delay) assuming a constant arrival rate λm.

In (4) and (7), the delay for the bursty system was normalized by the delay for the M/G/1 system. If instead we choose to normalize by the delay for an M/M/1 system with the same arrival and service rates (denoted by $\tilde{D}_m$), then the limits become

$$\lim_{\lambda\to 0} \frac{D_B}{\tilde{D}_M} = 1 + \frac{C^2-1}{2} + \frac{\mu}{\alpha}[1 - \phi(\beta)] \tag{8}$$

and

$$\lim_{\lambda \to \frac{\beta\mu}{\alpha}} \frac{D_B}{\hat{D}_M} = 1 + \frac{C^2 - 1}{2} + \frac{\mu}{\alpha}. \tag{9}$$

These results are interesting in that they suggest separation of the effects of variability of the service time $(C^2 - 1)/2$ and the variability of the arrival process (the third term).

In addition to the results for the mean delay D_B, we show that the light-traffic limit of the distribution of the delay D, given $D > 0$, is

$$\lim_{\lambda \to 0} P(D > t \mid D > 0) = \frac{\int_t^\infty \int_y^\infty [\alpha + \beta e^{-\beta(x-t)}] dH(x) dy}{\frac{\alpha}{\mu} + 1 - \int_0^\infty e^{-\beta x} dH(x)}. \tag{10}$$

Again, in order for this to be rigorously stated it should be in terms of the limit of similar quantities for systems allowing only a finite number of virtual calls.

III. DERIVATION OF THE LIGHT-TRAFFIC RESULTS

In this section, we derive the light-traffic result stated in (4). Our approach is to show that as the traffic intensity goes to 0, the probability of having i (greater than 0) packets in the system goes to 0 asymptotically as λ^i. The exact rate of convergence can be derived by a detailed study of the state equations and from there (4) follows trivially.

Consider a single-server queue whose service times are of phase type (see Appendix A). Let the arrival process be a nonhomogeneous Poisson process whose rate is λ times a function of the state of a Markov process. Then, the multidimensional process consisting of the number i of packets in queue, the state j of the arrival Markov process, and the phase k of the packet in service is itself a Markov process. A typical state will be denoted by (i, j, k) for $i > 0$ and $(0, j)$ for $i = 0$, where $j \geq 0$ and $k = 1, \cdots, m$, the number of phases. The ergodic distribution will be denoted by $\rho(\cdot, \cdot, \cdot)$ and define

$$\rho(i) = \sum_{j,k} \rho(i, j, k),$$

with the obvious definition for $i = 0$. When there is an upper bound on the arrival rate (as when the arrival Markov process is finite), then the technique used in Burman and Smith[10] to prove Theorem 3.1 therein can be employed to prove:

Lemma 1: There exists a constant $R > 0$ such that

$$\rho(i) \leq \lambda^i R^i.$$

One may define

$$\tilde{\rho}(i, j, k) = \lim_{\lambda \to 0} \lambda^{-1} \rho(i, j, k) \tag{11}$$

with an analogous definition for $i = 0$, and these limits may be recursively related and shown to exist by examination of the balance equations. (See Smith[11] and Burman and Smith[10] for examples of this technique.) Thus, when the arrival rates are bounded, the light-traffic methodology is straightforward. It is quite possible, however, that the resulting equations are difficult to solve.

This is exactly the case when the arrival Markov process is an M/M/N/N queue (finite number of virtual calls). It is difficult to explicitly solve for $\tilde{\rho}^{(N)}$ [the limiting light-traffic normalized probabilities for this system, see (11)], although it can be shown that $\tilde{\rho}^{(N)} \to \tilde{\rho}$ as $N \to \infty$, where $\tilde{\rho}$ is the solution to the equations (assuming Lemma 1) when the birth-death process is the M/M/∞ queue. At the core of this argument (not presented here in detail) are the facts that the equations involving $\tilde{\rho}^{(N)}(\cdot, j, \cdot)$ for $j < N$ are identical with those involving $\tilde{\rho}^{(N+1)}(\cdot, j, \cdot)$ and that

$$\lim_{N \to \infty} \tilde{\rho}^{(N)}(0, j) = \frac{1}{j!} (\alpha/\beta)^j e^{-\alpha/\beta}.$$

The existence of the limits [in (11)] can be shown by recursion on i and the fact that the limit was previously established for $i = 0$. Thus $\lim_{N \to \infty} \tilde{\rho}^{(N)} (= \tilde{\rho})$ may be calculated by studying the system with $N = \infty$.

We now turn our attention to calculating $\tilde{\rho}$ by studying the bursty system with arrival rate λ times the number of calls in an M/M/∞ queue. For this system it is not hard to show that ρ, the steady-state probability satisfies

$$\begin{aligned} &- (j\lambda + \alpha + j\beta)\rho(0, j) + \alpha\rho(0, j - 1) \\ &+ (j + 1)\beta\rho(0, j + 1) + \sum_n \rho(1, j, n)\mathrm{E}_n = 0, \\ &- (j\lambda + \alpha + j\beta - T_{kk})\rho(1, j, k) + \alpha\rho(1, j - 1, k) \\ &+ (j + 1)\beta\rho(1, j + 1, k) + \sum_n \rho(1, j, n)T_{nk} \end{aligned} \tag{12}$$

$$+ j\lambda\omega_k\rho(0, j) + \omega_k \sum_n \rho(2, j, n)\mathrm{E}_n = 0, \tag{13}$$

and

$$\begin{aligned} &- (j\lambda + \alpha + j\beta - T_{kk})\rho(i, j, k) + \alpha\rho(i, j - 1, k) \\ &+ (j + 1)\beta\rho(i, j + 1, k) + \sum_n \rho(i, j, n)T_{nk} \end{aligned} \tag{14}$$

$$+ j\lambda\rho(i - 1, j, k) + \omega_k \sum_n \rho(i + 1, j, n)\mathrm{E}_n = 0, \quad \text{for} \quad i \geq 2,$$

where $j \geq 0$ and $\omega\cdot$, $T\cdot\cdot$, and $\mathrm{E}\cdot$, are the initial, transition, and exit rates defining the phase-type distribution of the service-time process. (These are discussed in greater detail in Appendix A.) We also assume that Lemma 1 holds.

Next define the generating functions

$$q(i, z, k) = \sum_j z^j \tilde{\rho}(i, j, k), \qquad i > 0,$$

and

$$q(0, z) = \sum_j z^j \tilde{\rho}(0, j).$$

From (12) to (14), we see that q satisfies the following equations

$$-\alpha(1 - z)q(0, z) + \beta(1 - z)q_z(0, z) = 0, \tag{15}$$

$$q(1, z, \cdot)[T - \alpha(1 - z)I] + q_z(1, z, \cdot)\beta(1 - z)I = -\omega z q_z(0, z), \tag{16}$$

and

$$q(i, z, \cdot)[T - \alpha(1 - z)I] + q_z(i, z, \cdot)\beta(1 - z)I = -z q_z(i - 1, z, \cdot) \quad \text{for} \quad i > 1. \tag{17}$$

Equation (15) immediately gives

$$q(0, z) = e^{-\alpha/\beta(1-z)}. \tag{18}$$

The next lemma relates D_B, the expected delay in this system, to $q(1, z, \cdot)$.

Lemma 2:

$$\lim_{\lambda \to 0} \lambda^{-1} D_B = -\frac{\beta}{\alpha} q_z(1, 1, \cdot)T^{-1}e,$$

where e is the vector of ones.

Proof: The mean number L of packets in the queue is given by

$$L = \sum_{i>1} (i - 1) \sum_{j,k} \rho(i, j, k).$$

By Lemma 1,

$$\lim_{\lambda \to 0} \lambda^{-2} L = \sum_{j,k} \tilde{\rho}(2, j, k) = q(2, 1, \cdot)e.$$

From (17), we get that

$$q(2, 1, \cdot) = -q_z(1, 1, \cdot)T^{-1}$$

and by Little's Law we are done. □

We now establish (3). The previous lemma shows that the key

quantity is $q_z(1, 1, \cdot)$. Differentiating (16) with respect to z and evaluating at $z = 1$ gives

$$\alpha q(1, 1, \cdot)I + q_z(1, 1, \cdot)(T - \beta I) = \omega \frac{\alpha}{\beta}\left(\frac{\alpha}{\beta} + 1\right), \tag{19}$$

where we used (18) to give us $q(0, z)$. Substituting for $z = 1$ into (16) gives that

$$q(1, 1, \cdot) = \frac{\alpha}{\beta}(-\omega T^{-1}) = \frac{\alpha}{\beta\mu}\xi,$$

where ξ is the stationary distribution of the service-time process [see (42) and (43)]. Rearranging (19) we get

$$\begin{aligned} q_z(1, 1, \cdot) &= \frac{\alpha^2}{\beta}\omega T^{-1}(\beta I - T)^{-1} + \frac{\alpha}{\beta}\left(\frac{\alpha}{\beta} + 1\right)\omega(\beta I - T)^{-1} \\ &= -\frac{\alpha}{\beta^2}[\alpha\beta\omega T^{-1}(\beta I - T)^{-1} - (\alpha + \beta)\omega(\beta I - T)^{-1}] \\ &= -\frac{\alpha}{\beta^2}[\alpha\omega T^{-1} - \beta\omega(\beta I - T)^{-1}] \\ &= \left(\frac{\alpha}{\beta}\right)^2(-\omega T^{-1}) + \left(\frac{\alpha}{\beta}\right)\omega(\beta I - T)^{-1}. \end{aligned} \tag{20}$$

Finally, from Lemma 2 and Corollary 1, we get

$$\begin{aligned} \lim_{\lambda \to 0} \lambda^{-1} D_B &= -\frac{\beta}{\alpha} q_z(1, 1, \cdot)T^{-1}e \\ &= \frac{\alpha}{\beta}\frac{1}{\mu}(-\xi T^{-1}e) - \frac{1}{\mu}\xi T^{-1}(\beta I - T)^{-1}E \\ &= \frac{\alpha}{\beta}\frac{\mu_2}{2} + \int_0^\infty e^{-\beta x}\int_x^\infty H^c(s)ds dx. \end{aligned}$$

Normalizing by the expected delay in the M/G/1 queue and integrating by parts gives us (3). □

We next calculate the Laplace Transform $E(e^{-sD} \mid D > 0)$. In light traffic, a customer who is delayed ($D > 0$) will usually see only one customer in the system and will just wait for the service completion. This is made rigorous by Lemma 1. The probability that such a customer arrives and finds the server in phase k is given by

$$\frac{\sum_m m\rho(1, m, k)}{\sum_{m,n} m\rho(1, m, n)},$$

or in terms of q this is

$$\frac{q_z(1, 1, k)}{\sum_n q_z(1, 1, n)} . \tag{21}$$

From (21) we get that

$$E(e^{-sD} \mid D > 0) = \frac{\sum_k q_z(1, 1, k) E_k(e^{-s\tau})}{\sum_n q_z(1, 1, n)}$$

$$= \frac{q_z(1, 1, \cdot)(sI - T)^{-1}\mathrm{E}}{q_z(1, 1, \cdot)e} , \tag{22}$$

where $E_k(e^{-s\tau})$ is the Laplace transform of the remaining service time given that the current phase is k, and we have used proposition (1) from Appendix A.

Evaluating the denominator first, from (20) and Corollary 1 we see that

$$[q_z(1, 1, \cdot), e] = \left(\frac{\alpha}{\beta}\right)^2 \frac{1}{\mu} (\xi, e) + \left(\frac{\alpha}{\beta}\right) \frac{1}{\mu} \xi(BI - T)^{-1}\mathrm{E}$$

$$= \left(\frac{\alpha}{\beta}\right)^2 \frac{1}{\mu} + \left(\frac{\alpha}{\beta}\right) \int_0^\infty e^{-\beta x} H^c(x) dx$$

$$= \left(\frac{\alpha}{\beta}\right)^2 \frac{1}{\mu} + \left(\frac{\alpha}{\beta}\right) \frac{1 - \tilde{H}(\beta)}{\beta} ,$$

where ξ is the vector of ergodic probabilities for the service-time variable and $\tilde{H}(s) = \int e^{-sx} dH(x)$ is the Laplace Transform of the service-time density. The numerator, after some algebraic reduction, becomes

$$q_z(1, 1, \cdot)(sI - T)^{-1}\mathrm{E} = \left(\frac{\alpha}{\beta}\right)^2 \omega[(-T)^{-1}(sI - T)^{-1})\mathrm{E}$$

$$+ \left(\frac{\alpha}{\beta}\right) \frac{\omega(\beta I - T)^{-1}\mathrm{E} - \omega(sI - T)^{-1}\mathrm{E}}{s - \beta}$$

$$= \left(\frac{\alpha}{\beta}\right)^2 \frac{1 - \tilde{H}(s)}{s} + \left(\frac{\alpha}{\beta}\right) \frac{\tilde{H}(\beta) - \tilde{H}(s)}{s - \beta} .$$

Combining the two calculations, we get

$$E(e^{-sD} \mid D > 0) = \frac{\alpha \left[\frac{1 - H(s)}{s}\right] + \frac{\beta}{s - \beta} [\tilde{H}(\beta) - \tilde{H}(s)]}{\alpha \frac{1}{\mu} + 1 - \tilde{H}(\beta)} . \tag{23}$$

It is not difficult to show that (23) is the transform of (10).

IV. DERIVATION OF THE HEAVY-TRAFFIC RESULTS

In this section we describe the technique that allows us to arrive at (5) and (6). This approximation is derived by first proving that the number of packets in the system, when appropriately scaled, converges to a diffusion process $X(t)$ as $\rho^{(N)}$ converges to one. Given the drift and infinitesimal variance of the diffusion process, one can obtain the steady-state mean of $X(t)$, and by using Little's Theorem, we get the approximation given in (5).

The main theorem of this section, which can be proved rigorously using the techniques in Burman,[12] forms the theoretical basis for (6). We will outline its proof and show in detail how to calculate the mean and variance of the resulting diffusion, which from an application point of view is the difficult part of developing these approximations.

Consider a sequence of processes $X_n(t)$, where the nth process represents the number of packets at time t in a single-server system to which packets arrive at an instantaneous rate λ_n times the state of an M/M/N/N queueing system (representing the number of virtual calls). The heavy-traffic limits are found by defining λ_n as follows:

$$\lambda_n E^{(N)}V = \mu - \frac{\delta}{\sqrt{n}}, \tag{24}$$

where δ is a positive constant and $E^{(N)}V$ is the expected number of calls in the M/M/N/N system. We are initially interested in studying the sequence of scaled processes $n^{-1/2}X_n(nt)$ as $n \to \infty$.

We start by investigating the limiting behavior of the infinitesimal generators of a sequence of Markov processes created by appending supplementary variables to $X_n(t)$. A limiting generator is identified and the Trotter-Kato Theorem (see Kato[13]) implies that the finite-dimensional distributions converge. The form of the limiting generator completely determines the limiting process. Weak convergence can be established via a theorem of Stroock and Varadhan.[14]

Let $V(t)$ be the number of virtual calls at time t and let $Y(t)$ be the time since the packet currently in service entered service. The multi-dimensional process

$$M_n(t) = [n^{-1/2}X_n(nt),\ Y(nt),\ V(nt)]$$

is Markov. We denote a typical element of the state-space as (x, y, j), where $x = 0, n^{-1/2}, 2n^{-1/2}, \cdots, y \in [0, \infty)$ and $j = \{0, 1, \cdots, N\}$. For the remainder of this paper, let $h(y)$ be the density of the service-time distribution,

$$H^c(y) = \int_y^\infty h(s)ds, \qquad \mu(y) = \frac{h(y)}{H^c(y)}$$

and assume that for some constants γ and Γ,

$$0 < \gamma \leq \mu(y) \leq \Gamma \leq \infty. \tag{25}$$

The implications of and conditions for relaxing (25) are discussed in Burman.[12] These conditions are required for the method of proof and do not pose any significant restrictions on the results.

Let B and Q be the infinitesimal generators of the processes $V(t)$ and $Y(t)$, respectively. The generator B is an $(N+1) \times (N+1)$ matrix given by

$$B = \begin{bmatrix} -\alpha & \alpha & & & & \\ \beta & -(\beta+\alpha) & \alpha & & & \\ & 2\beta & -(2\beta+\alpha) & \alpha & & \\ & & & \ddots & & \\ & & & & N\beta & -N\beta \end{bmatrix}$$

and Q is a first-order linear differential operator given by

$$Qf(y) = f'(y) + \mu(y)[f(0) - f(y)].$$

The following two lemmas are needed to carry out the calculations needed in the main theorem of the section (to be stated below):

Lemma 3: The equation

$$Bv = w \tag{26}$$

has a solution v if and only if

$$(e, w) = 0, \tag{27}$$

where $e_k = 1/k!(\alpha/\beta)^k = 1/k!\ \rho^k$. In this case v is unique to an additive constant and is given by

$$v(k+1) = v(0) + \frac{1}{\alpha} \sum_{j=0}^{k} \frac{j!}{\rho^j} \sum_{i=0}^{i} \frac{\rho^i}{i!} w(i). \tag{28}$$

Proof: Since B is a finite-dimensional linear operator corresponding to an ergodic Markov process, it has a one-dimensional null space spanned by the vector e. Hence, (27) is both necessary and sufficient for a unique solution of (26) to exist. This solution is easily seen to be given by (28). □

Lemma 4: The equation

$$Qf = g \tag{29}$$

has a solution f if and only if

$$\int_0^\infty g(y)H^c(y)dy = 0. \tag{30}$$

In this case, f is unique to an additive constant and is given by

$$f(y) = \frac{1}{H^c(y)} \int_0^y H^c(s)g(s)ds. \tag{31}$$

Proof: The operator Q is the generator of an ergodic Markov process with the stationary density $\mu H^c(y)$. A unique solution exists if and only if (30) holds. Solving the first-order linear differential equation gives (31). □

We are now ready to state the main theorem of this section:

Theorem 1: Under assumption (25),

$$n^{-1/2}X_n(nt) \Rightarrow X(t),$$

where $X(t)$ is a diffusion on R^+ with pure reflection at the origin, downward drift δ and infinitesimal variance s^2 given by

$$s^2 = \frac{1}{2}\left\{\lambda E^{(N)}V + \mu^3\sigma^2 + \sum_{k=0}^{N} \frac{1}{\alpha P_k}\left[\sum_{j=0}^{k-1} P_j(\mu - j)\right]^2\right\}, \tag{32}$$

where

$$P_k = \frac{1}{k!}\left(\frac{\alpha}{\beta}\right)^k \Big/ \sum_{j=0}^{N} \frac{1}{j!}\left(\frac{\alpha}{\beta}\right)^j$$

is the steady-state probability of finding k calls in the $M/M/N/N$ blocking system.

Proof of Theorem: As mentioned earlier, we will show how to identify the mean and variance of (32). The infinitesimal generator for $M_n(t)$ is given by

$$A_nf(x, y, j) = n(B + Q)f + \lambda j\left[f\left(x + \frac{1}{\sqrt{n}}, y, j\right) - f(x, y, j)\right]$$
$$+ \mu(y)\left[f\left(x - \frac{1}{\sqrt{n}}, 0, j\right) - f(x, 0, j)\right] \quad \text{for} \quad x \geq \frac{1}{\sqrt{n}}, \tag{33}$$

and

$$A_nf(0, 0, j) = nBf + \lambda j\left[f\left(\frac{1}{\sqrt{n}}, 0, j\right) - f(0, 0, j)\right]. \tag{34}$$

For $f(x)$ twice continuously differentiable with $f'(0) = 0$, set

$$f_n(x, y, j) = f(x) + \frac{1}{\sqrt{n}} f'(x)g(y, j) + \frac{1}{n} f''(x)h(y, j). \tag{35}$$

We construct bounded functions g and h so that

$$A_nf_n \to Af$$

uniformly, where

$$Af(x) = mf'(x) + s^2 f''(x).$$

The constants m and s^2 are determined in the process. Given (25), these constructions will complete the proof (see Burman[12]).

From (35) and (33) we see that

$$\begin{aligned} A_n f_n = {} & n(B + Q)f + \sqrt{n} f'(x)[\lambda_n j - \mu(y) + (B + Q)g] \\ & + f''(x)\left\{\frac{1}{2}[\lambda_n j + \mu(y)] + \lambda_n j g(y, j) - \mu(y) g(0, j)\right. \\ & \left. + (B + Q)h(y, j)\right\} + 0\left(\frac{1}{\sqrt{n}}\right). \end{aligned} \tag{36}$$

First note that since f is independent of y and j, $(B + Q)f = 0$. Set g equal to the solution of the equation

$$(B + Q)g = -[\lambda_n j - \lambda_n E^{(N)}V - \mu(y) + \mu]. \tag{37}$$

Lemmas 3 and 4 imply that g exists and

$$\begin{aligned} g(y, j) = {} & \frac{1}{H^c(y)} \int_0^y H^c(s)[\mu(s) - \mu] ds \\ & - \frac{\lambda_n}{\alpha} \sum_{k=0}^{j-1} \frac{k!}{\rho^k} \sum_{i=0}^{k} \frac{\rho^i}{i!} [j - E^{(N)}V]. \end{aligned} \tag{38}$$

Let h be the solution of the equation

$$(Q + B)h = -[w(y, j) - s^2], \tag{39}$$

where $w(y, j) = 1/2[\lambda_n j + \mu(y)] + \lambda_n j g(y, j) - \mu(y) g(0, j)$. The constant s^2 [see (26)] can be easily calculated from conditions (27) and (30), i.e.,

$$\sum_{i=0}^{N} \frac{\rho^i}{i!} \int_0^\infty H^c(y) w(y, i) dy = 0.$$

Using g and h calculated above we have

$$A_n f_n = -\delta f'(x) + s^2 f''(x) + 0\left(\frac{1}{\sqrt{n}}\right).$$

When $x = 0$ analogous calculations give that $A_n f_n$ converges provided $f'(0) = 0$, and the proof is complete. □

The limit theorem for the M/G/1 queue with arrival rate $\lambda_n E^{(N)}V$ (see Iglehart and Whitt[15]) gives the same mean δ with the variance $\lambda E^{(N)}V + \mu^3 \mu_2$ for the limiting diffusion. Calculating the steady-state expected number in the limiting systems and using Little's Theorem

(see Cooper[16]) yields the asymptotic expression for the ratio of $D_B^{(N)}$ and $D_M^{(N)}$, the delays in the two systems:

$$\frac{D_B^{(N)}}{D_M^{(N)}} \approx 1 + 2\,\frac{\sum_{k=0}^{N-1} \frac{1}{P_k} \left[\sum_{i=0}^{k} P_i(\mu - \lambda i)\right]^2}{\alpha\mu^3\mu_2}. \tag{40}$$

Letting N go to infinity, P_k goes to the Poisson distribution and (40) goes to

$$1 + 2\mu/\alpha\left(\frac{1}{1 + C^2}\right)$$

as promised. □

Remark: We are careful here not to write (40) as a limit but as an approximation. In order to strictly prove convergence in (40), two rather technical steps remain to be proved. The first is that the steady-state number in the queue (when scaled) converges to the steady-state value of the diffusion. The second is that the expected values of these steady states converge. Both are difficult issues in themselves and, aside from mathematical completeness, add little information to the approximation, which is the germaine issue of this paper.

V. CONCLUDING REMARKS

We considered here a single-server queueing system with a nonhomogeneous Poisson arrival process whose rate is some constant λ times the state of an independent M/M/∞ system. In this paper, we derived limiting values for the mean delays as the traffic intensity goes to zero and to one.

This system was used to model the delay of packets at a packet switch. In this setting, an individual data customer generates packets at a constant rate λ and the number of customers generating packets is given by the state of the M/M/∞ system. This is, of course, not the only possible model for packet arrivals. Packets generated by an individual virtual call may be "smoother" or more "bursty" than packets generated by a Poisson process. It is also possible that restrictions could be imposed on the total number of simultaneous virtual calls, thereby making the infinite-server assumption unrealistic. Work is currently under way to extend the techniques of this paper to cover these and other more general models of bursty traffic.

Our ultimate goal in studying D_B/D_M in both light and heavy traffic is to be able to derive simple, closed-form approximations for D_B for all values of the traffic intensity ρ. One candidate approximation is to

linearly interpolate between the values obtained for D_B/B_M when $\rho \to 0$ and when $\rho \to 1$ [see (4) and (5)]. This approach gives the following estimate for the mean delay:

$$D_B \sim \frac{\rho}{1-\rho}\frac{(1+C^2)}{2\mu}\left[1+\frac{\mu}{\alpha}\frac{1-(1-\rho)\phi(\beta)}{(1+C^2)}\right].$$

More sophisticated interpolations are possible (see Lauber and Smith[7] for applications of these techniques to estimating the delays in an M/G/c queue); however, in the absence of any test data, verification is difficult. One method of verification is to study the accuracy of these approximation techniques when applied to other models of bursty arrivals for which explicit expressions are known (see Yechiali and Naor[17]). This work is currently under way and will be reported on in the future.

REFERENCES

1. H. Heffes, "A Class of Data Traffic Processes—Covariance Function Characterization and Related Queueing Results," B.S.T.J., *59*, No. 6 (July–August 1980), pp. 897–929.
2. A. Kuczura, "Queues with Mixed Renewal and Poisson Input," B.S.T.J., *51*, No. 6 (July–August 1972), pp. 1305–26.
3. R. V. Laue, unpublished work.
4. M. F. Neuts, "A Versatile Markovian Point Process," J. Appl. Prob., *16* (December 1979), pp. 764–79.
5. D. M. Lucantoni and M. F. Neuts, "Numerical Methods for a Class of Markov Chains Arising in Queueing Theory," Technical Report No. 78/10 (May 1978), Department of Statistics and Computer Science, University of Delaware.
6. D. Anick, D. Mitra, and M. M. Sondhi, unpublished work.
7. P. J. Lauber and D. R. Smith, unpublished work.
8. D. Mitra and D. Anick, unpublished work.
9. L. Kosten, "Stochastic Theory of a Multi-Entry Buffer (1)," Delft Progress Report, *1*, 10–18, 1974.
10. D. Y. Burman and D. R. Smith, "A Light Traffic Theorem for Multi-Server Queues," Math. Oper. Res., *8*, No. 1 (February 1983), pp. 15–25.
11. D. R. Smith, "Optimal Repairman Allocation—Asymptotic Results," Management Science, *24*, No. 6 (February 1978), pp. 665–74.
12. D. Y. Burman, unpublished work.
13. T. Kato, *Perturbation Theory for Linear Operators*, Berlin: Springer-Verlag, 1976.
14. D. W. Stroock and S. R. S. Varadhan, *Multidimensional Diffusion Processes*, Berlin: Springer-Verlag, 1979.
15. D. L. Igelhart and W. Whitt, "Multiple Channel Queues in Heavy Traffic," I. *Advances in Applied Probability*, *2*, No. 1 (Spring 1970), pp. 150–77.
16. R. Cooper, *Introduction to Queueing Theory*, New York: North Holland, 1981.
17. V. Yechiali and P. Naor, "Queueing Problems with Heterogeneous Arrivals and Service," Oper. Res., *19*, No. 3 (May–June 1971), pp. 722–34.
18. J. Keilson, *Markov Chain Models—Rarity and Exponentiality*, New York: Springer-Verlag, 1979.

APPENDIX A

Background on Phase-Type Distributions

Appendix A summarizes results for phase-type distributions that are used in Section III. A service time is said to have phase-type distribution if it is distributed like the first exit time from a continu-

ous-time, finite-state Markov chain. Any distribution can be arbitrarily approximated in the sense of weak convergence by one of phase type. We assume that the states (also known as phases) are the nonnegative integers i, $i = 1, \cdots, m$. Let the rate of transition from phase i to phase j be T_{ij} $(i \neq j)$, and the rate of exiting from phase i be ξ_i. Define $T_{ii} = -\sum_{j \neq i} T_{ij} - \mathrm{E}_i$, i.e., minus the rate of leaving state i. We assume that T, the matrix of T_{ij}, is invertible; this is sufficient to imply that the real part of the spectrum of T is strictly negative. A customer starts service in phase i with probability ω_i. Following Neuts,[4] we use the notation (ω, T) to describe this distribution, where ω is the vector of initial probabilities and T is the m-dimensional matrix of rates.

The vector ξ of ergodic probabilities for the service phase of the renewal process is determined by normalizing the solution to the equations

$$-\xi_i T_{ii} = \sum_j \xi_j T_{ji} + (\Sigma \xi_j \mathrm{E}_j)\omega_i, \tag{41}$$

or in matrix notation,

$$\xi T = -(\xi, \mathrm{E})\omega. \tag{42}$$

The service rate μ is defined as

$$\mu = (\xi, \mathrm{E}). \tag{43}$$

It is not difficult to show that μ is the reciprocal of the mean service time.

Let τ be the first exit time from the chain; let $\mathrm{M}_i(s) = E_i(e^{-s\tau})$, the Laplace Transform of τ given the initial state is i; and $M_i^n = E_i \tau^n$. The results of Proposition 1 are well known (see Kielson,[18] page 82, and Burman and Smith[12]).

Proposition 1. For τ, $\mathrm{M}(s)$, and μ^n defined above,

$$\mathrm{M(s)} = (\mathrm{sI} - \mathrm{T})^{-1}\mathrm{E} \tag{44}$$

and

$$M^n = n!(-1)^n T^{-n} e, \tag{45}$$

where e is the vector of all ones. In particular, the nth moment of the service-time distribution $(\tilde{M}^n)$ is

$$\tilde{M}^n = n!(-1)^n(\omega, T^{-n}e). \tag{46}$$

If $H^c(x)$ equals the tail of the service-time distribution, then the following corollary is immediate.

Corollary 1.

$$-\xi T^{-1}e = \int_0^\infty x\mu H^c(x)dx = \frac{\mu\mu_2}{2}, \tag{47}$$

and

$$-\xi T^{-1}(\beta I - T)^{-1}\mathrm{E} = \int_0^\infty e^{-\beta x}\int_x^\infty \mu H^c(s)dsdx. \tag{48}$$

Proof: To see (47), observe that $\mu H^c(x)$ is the density of the remaining service time when the process is sampled in equilibrium, and apply Proposition 1. The identity on the integral can be obtained by integration by parts. In a similar fashion,

$$\xi T^{-1}(T - \beta I)^{-1}\mathrm{E} = \frac{\xi[(T - \beta I)^{-1} - T^{-1}]\mathrm{E}}{\beta}$$

$$= \frac{\left[\int_0^\infty \mu H^c(x)dx - \int_0^\infty e^{-\beta x}\mu H^c(x)dx\right]}{\beta},$$

and integration by parts gives us (48). □

APPENDIX B

Intuitive Derivation of the Results in Light Traffic

The following argument is based on the intuitive notion that, in light traffic, almost all arriving packets arrive to an empty system and are served before the arrival of another packet. Furthermore, the times when there is exactly one packet in the system constitute almost all of the time in which an arriving packet might be subject to delay. Thus, in light traffic, it is easy to derive the proportion of time in which an arrival will be delayed, and to find the delay for such an arrival. A slight complication is introduced in the analysis by the fact that the arrival rate of packets is not constant, but this is easily overcome.

While the overall derivation is not rigorous, it is convincing and gives insight into the light-traffic behavior. The results, of course, are consistent with the rigorous results derived for phase-type distributions in Section III.

We will begin with traffic-independent results for the model described previously in the beginning of Section II, in which the instantaneous arrival rate equals λ times the state of an M/M/∞ queue with birth rate α and service rate equal to β. The equilibrium distribution of the M/M/∞ queue governing the arrival process is easily seen to be Poisson with mean α/β. However, this is not its distribution immediately after an arrival, since the rate of arrivals is proportional to the

state of the M/M/∞ queue. Thus, the conditional probability that the state of the M/M/∞ queue is m (for $m \geq 1$) immediately after an arrival is

$$\frac{m(\alpha/\beta)^m \frac{1}{m!} e^{-\alpha/\beta}}{\sum_{k=0}^{\infty} k(\alpha/\beta)^k \frac{1}{k!} e^{-\alpha/\beta}} = \frac{1}{(m-1)!} (\alpha/\beta)^{m-1} e^{-\alpha/\beta}.$$

This may be thought of as the distribution of a Poisson random variable with mean (α/β) with support right-shifted one unit. This point of view is useful in computing the distribution of the state of the M/M/∞ queue t time units after an arrival of a packet (denoted N_t), since the Poisson distribution is stationary, and the additional unit is still present with probability $e^{-\beta t}$. Thus,

$$E(z^{N_t}) = e^{(\alpha/\beta)(z-1)}[1 + e^{-\beta t}(z-1)]. \tag{49}$$

Now consider the system in light traffic. Denote the state of the system by (N, M, T) where N is the number of packets in the system, M is the state of the M/M/∞ queue that modulates the arrival process, and T is the remaining service time of the packet being served.

We first obtain an expression for $E(e^{-sT}z^M\delta_{1N})$ in light traffic, where $\delta_{1N} = 0$ for $N \neq 1$ and $\delta_{11} = 1$. We then show how to use this quantity to obtain the desired limits. In light traffic, almost every packet arrives to an empty system and is served before another arrival. By Little's Law

$$P(N = 1) = E\delta_{1N} \sim \lambda\alpha E(S)/\beta.$$

Again, making the assumption of the last sentence and looking at the system at only those times for which $N = 1$, we obtain by standard renewal theory arguments that

$$E(e^{-sT}z^M N = 1) = \frac{1}{ES} E\left[\int_0^S E(z^{N_t})e^{-s(S-t)}dt\right] \stackrel{\text{def}}{=} \frac{1}{ES} G(z, s).$$

Multiplying, we obtain that

$$E(e^{-sT}z^M\delta_{1N}) \sim \lambda(\alpha/\beta)G(z, s). \tag{50}$$

By conditioning on S and using (49), we obtain

$$G(z, s) = e^{(\alpha/\beta)(z-1)} \frac{1}{s}\left\{[1 - \tilde{H}(s)] + \left(\frac{z-1}{\beta - s}\right)[\tilde{H}(s) - \tilde{H}(\beta)]\right\}, \tag{51}$$

where $\tilde{H}$ is the Laplace Transform of a service time.

Equations (50) and (51) can be used to find desired properties of the queue in light traffic. For example, since nearly all packets that

are delayed are generated when there is only one other packet in the system, and since the rate of generation of packets is λ times the number of calls, we find the rate of generation of delayed packets, λ_D, is

$$\lambda_D \sim \lambda^2(\alpha/\beta)G_z(1, 0), \tag{52}$$

or

$$\lambda_D \sim \lambda^2 \left\{(\alpha/\beta)^2 \frac{1}{\mu} + \alpha/\beta[1 - \tilde{H}(\beta)]\right\}. \tag{53}$$

The Laplace Transform of the delay, given that the delay is greater than 0, defined to be $\phi(s)$, is also found easily since the delay of a packet equals the remaining service time on arrival. Thus,

$$\phi(s) \sim \frac{G_z(1, s)}{G_z(1, 0)},$$

or

$$\phi(s) \sim \frac{\alpha/s[1 - \tilde{H}(s)] + \dfrac{\beta}{\beta - s}[\tilde{H}(s) - \tilde{H}(\beta)]}{\alpha/\mu + 1 - \tilde{H}(\beta)}. \tag{54}$$

Inversion of the Laplace Transform gives (10) and also gives

$$E(D \mid D > 0) \approx \frac{\dfrac{\alpha\mu_2}{2} + \dfrac{1}{\mu} - \dfrac{1}{\beta}[1 - \tilde{H}(\beta)]}{\alpha/\mu + 1 - \tilde{H}(\beta)}, \tag{55}$$

where $\mu_2 = E(S^2)$. Of course, $P(D > 0) = \dfrac{\lambda_D}{\lambda(\alpha/\beta)}$ so that (52) and (54) give

$$E(D) \approx \frac{\lambda}{\beta}\left\{\frac{\alpha\mu_2}{2} + \frac{1}{\mu} - \frac{1}{\beta}[1 - \tilde{H}(\beta)]\right\}. \tag{56}$$

This may be combined with the Pollaczek-Khintchine formula to give (3).

AUTHORS

Donald R. Smith, A.B. (Physics), 1969, Cornell University; M.S. (Operations Research), 1974, Columbia University; Ph.D. (Operations Research), 1975, University of California, Berkeley; Bell Laboratories, 1980—. Before joining Bell Laboratories, Mr. Smith was employed at Adaptive Technology, Inc. from 1970 to 1974, and was Assistant Professor in the Department of Industrial Engineering and Operations Research, Columbia University from 1975 to 1979. At Adaptive Technology, Mr. Smith developed mathematical models for

new techniques in statistical multiplexing. At Bell Laboratories he is in the Operations Research Department of the Network Analysis Center pursuing interests in applied stochastic processes.

David Y. Burman, B.S. (Mathematics), 1968, C.C.N.Y.; Ph.D. (Applied Mathematics), 1979, New York University; Bell Laboratories, 1969—. At Bell Laboratories, Mr. Burman has worked on problems in network flows, scheduling, and stochastic processes.

THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 6, July-August 1983
Printed in U.S.A.

Modernization of the Suburban ESS:

Overview: Evolution of the Suburban ESS

By T. E. GRASSMAN* and J. E. YATES*

(Manuscript received June 8, 1982)

This paper highlights the systematic evolution of the No. 2/2B Electronic Switching System (ESS), describing both hardware and software changes. It also introduces the subsequent articles that describe the modern interfaces and modern development environment enjoyed by the system. The No. 2 ESS was the first electronic switching system specifically designed for the suburban community. Subsequent incorporation of a more flexible and powerful processor provided a basis for a large increase in processing capacity and a continuing modernization of system structure. The evolved system, the No. 2B ESS, is now rich in features and offers modern administrative and maintenance capabilities.

I. INTRODUCTION

The No. 2 Electronic Switching System (ESS),[1] designed in the 1960s to provide ESS capabilities to suburban communities, was developed under the same constraints as other large real-time control systems of that era. Memory was expensive and processors were not powerful enough to compensate for inefficiencies in software. As a

* Bell Laboratories.

result, programs were written in assembly language and were tightly coupled. Compared with No. 1 ESS, then being deployed in urban areas, the No. 2 ESS faced a more stringent cost sensitivity due to the smaller number of subscribers supporting a given local switching system.

The original No. 2 Processor was replaced in 1976 with the 2B Processor, which has a more modern architecture based on improved semiconductor technology. This more powerful processor provided capacity for feature enhancement[2] and allowed utilization of more generalized and structured software techniques that have subsequently been incorporated into the system. This incorporation was accomplished by strictly controlling the portions of the software system that were allowed to change. Since that time the programs have continued to evolve and exploit the capabilities of the 2B Processor.

Most recently, the inclusion of modern software constructs in the 2BE3 generic has produced a design containing a modern data link interface that supports such capabilities as the 10A Remote Switching System (RSS), Automatic Message Accounting (AMA) data teleprocessing, and traffic data teleprocessing. Description of these features may be found in the companion articles of this series.

The evolutionary approach used to realize the present No. 2B ESS has resulted in a continuously viable product that remains economically attractive and modern in feature content. This paper will discuss the system evolution, and serve as an introduction to the other papers in this series.

II. HARDWARE AND SYSTEM CHARACTERISTICS

2.1 No. 2 ESS

The original No. 2 ESS central processor was designed using Transistor Resistor Logic (TRL) in the early 1960s. The architecture consisted of special purpose registers that were designed to accommodate a set of instructions tailored for use in telephone switching. For example, explicit instructions were included to support the software structure that used 8-word control blocks to process each telephone call. Its architecture and instruction set gave the No. 2 ESS the capability to efficiently switch telephone calls in a small to medium (2,000- to 20,000-line) office.

The hardware architecture used in the No. 2 ESS Processor supported two types of stores (memories). One of the stores was a permanent magnet twistor that contained the program as well as the data which specified the characteristics of lines and trunks. This 22-bit-wide memory was magnetically alterable off-line. The memory organization consisted of four 64K modules for a maximum of 256K

words. The selection of a module was independent of the normal addressing mode within a module. This required each program to independently administer the module selection for itself. The other store contained up to 32K 16-bit words and was on-line writable. It contained dynamic information related to the processing of calls.

An additional characteristic of No. 2 ESS was a wired logic unit that autonomously scanned for customer requests, and, under timed control, scanned for short-interval signals. Both of these logic control items deloaded the program and increased the efficiency of the overall system. Scanners and other peripheral units were controlled by a parallel bus system.

This processor was put into service in 1970 in the No. 2 ESS system and provided the desired service level, features, and cost benefits at the time of its initial cutover.

2.2 The No. 2B ESS

2.2.1 Background

In 1971, a series of cost reduction studies led to the selection of the more powerful 3A Processor Control Complex (3A CC)[3] for incorporation into the No. 2 ESS. The 3A CC was designed for use in both No. 2B ESS and No. 3 ESS. The 3A CC was designed along general-purpose computer principles rather than on the specific switching logic instructions incorporated in the original No. 2 ESS Processor, and made extensive use of integrated circuits. The new processor could address up to one million words of relatively inexpensive semiconductor memory, provided a general-purpose instruction set and a serial bus access to peripheral units, and used a microprogram-based instruction decoder.

The principal asset of the 3A CC Processor was its microprogrammed instruction set. This feature allowed the processor to not only encompass the No. 3 ESS commands, but also to be extended in a relatively simple manner to emulate the original No. 2 ESS instruction set, and to add additional instructions that were parallel in nature to the No. 2 ESS set. This allowed the retention of most of the programs that had previously been developed to provide features and maintenance on the existing No. 2 ESS (except for the processor maintenance itself).

To minimize changes in the control of major peripheral equipment, a new interface [the 2B Input/Output (I/O) unit] was developed to connect the 3A CC to the parallel peripheral bus system. (The combination of the 3A CC, the 2B I/O unit, the increased memory, and the microcode for emulation is referred to as the 2B Processor.) In addition, programs already developed for the 3A CC, e.g., teletype-

writer (TTY) handlers and processor maintenance software, were adopted intact to avoid duplicate software development.

2.2.2 Instruction emulation

The key element in replacing the No. 2 Processor was the emulation of the existing No. 2 ESS instruction set. The existing 22-bit data and instruction format of the No. 2 ESS did not fit in the 16-bit 3A CC memory. Converting each No. 2 ESS word to two 3A CC words would have resulted in an addressing incompatibility between the old and new systems, i.e., the same instruction would exist at different addresses in the two systems. If this were allowed to occur, object-level changes to correct field problems would have to be developed twice, once for the No. 2 ESS and again for the No. 2B ESS. By keeping the object-level addresses synchronized, this duplicate effort was avoided. The decision was made to expand the 3A CC store to 24 usable bits to allow exact emulation of No. 2 ESS instructions and data. Half-word instruction pairs could be accommodated in the same addressable word, and data packing was then identical in the two processors. A further decision was made to use the No. 2 ESS instruction manual as the requirements for the emulated instructions and to map the functional No. 2 ESS registers into the general-purpose registers of the 3A CC. Each register and instruction was fully emulated.

These restrictions gave several distinct advantages although they did not allow improvements in program techniques through use of the richer 3A CC instruction set. The advantages were as follows: a test program that previously tested the instruction logic on No. 2 ESS provided the acceptance test for the microprogrammed instruction set (i.e., no new program was required); an immediate check on correct program compilation was available by comparing raw program sizes, symbol values, and linkage characteristics between the existing and the emulated programs; debugging of these programs was required only on one machine (No. 2 or 2B Processor) since they could be automatically assembled from the same source for the other machine without human interference (this allowed nearly simultaneous program releases on the two different systems); and most existing documentation and procedures for the No. 2 ESS were usable with the 3A CC since most user interfaces were identical between the systems.

With these constraints, the bulk of No. 2 ESS programs (85 percent) could be used as they were currently written. Most of the remaining 15 percent were No. 2 ESS Processor maintenance programs, which were replaced in total by the 3A CC Processor maintenance programs. The programming effort of the 2B introduction then was completely restricted to the support programs (assemblers, loaders, etc.), six

interface programs, and new diagnostics and recovery programs for the wide store.

2.2.3 Input/output emulation

The No. 2 ESS wired logic I/O performed two functions. It looked for new customer service requests and collected the digits as they were dialed by the customer. In the new system, software was used to scan for service requests. The digit collection functions were exactly duplicated by an interrupt-level microcode sequence on the 3A CC. This provided an unchanged interface to existing programs that had depended on the wired logic. The only associated changes required were in recovery programs for handling digit collection failures.

The interface between the 3A CC and the No. 2 peripheral equipment was provided by the 2B I/O unit. This was implemented to be driven by the existing 3A CC serial channel interface. Driver programs were implemented in microcode so that controlling programs, issuing I/O instructions, were not required to change.

2.3 No. 2B ESS field introduction and evolution

The No. 2B ESS was successfully put into service in 1976. Less than one year later the same program was used to provide service in an office requiring increased capacity through a retrofit of the processors. Since the 3A CC is much faster than the No. 2 ESS Processor, the No. 2B ESS call capacity (35,000 calls/hour engineered load) is much greater than that of the No. 2 ESS (16,500 calls/hour engineered load). By 1977, No. 2 ESS Processors were no longer being manufactured.

The No. 2/2B ESSs continued to evolve gradually—and in step—for several years while their programs were kept locked together by the restrictions discussed above. As the number of in-service No. 2 ESSs began to decline through processor retrofits, and the pressure to provide new and expanded features increased, a decision was made to discontinue new development on the No. 2 ESS, and to take further advantage of the 3A CC Processor's capabilities in No. 2B ESS.

The 2BE3 generic program, put into service in November of 1981, utilizes significant software restructuring in certain areas. The resulting advantages range from achieving expanded capacities for per-line features (e.g., call forwarding) to the development of several major new features and the communications interface that supports them.

2.4 A modern data communications interface

The need for data communications between local switching offices and other systems (remote switches and various operation support

systems) has expanded in recent years. To meet this increasingly important need, the No. 2B ESS has incorporated, in the 2BE3 generic program, a modern data communications interface capability. This interface is based on the Serial Peripheral Unit Controller/Data Link (SPUC/DL), a cost-effective and flexible hardware element utilizing microprocessor control capable of supporting various communications protocols. The 3A CC serial channel I/O capability provides the high-speed communication path between the host and the SPUC/DL. An accompanying article, "A New Data Link Controller," describes the SPUC/DL and its architecture, design, and capabilities in considerable detail.

Firmware has been developed for the SPUC/DL to support levels 1 and 2 of the BX.25 data communications protocol. This capability is utilized in provision of interfaces to the Automatic Message Accounting Collection System (AMACS) and to the Engineering and Administrative Data Acquisition System (EADAS). Firmware has also been developed to support the RSS protocol, which predated the BX.25 standard, enabling the No. 2B ESS to provide control of the 10A RSS. The features associated with these interfaces are also described later in this series.

III. NO. 2B ESS SOFTWARE EVOLUTION

Soon after the No. 2B ESS was successfully cut into service, an effort was begun to exploit the capabilities of the new processor. This effort, which had been planned from the beginning, aimed at eliminating data and program constraints. The success of these initial enhancements led to other incremental improvements. The program has matured to the point that enhancements driven by modern software practices, as well as the 2B Processor's architecture, are being pursued. These include operating-system-type primitives, data independence, and program decoupling.

This section will review some enhancements that demonstrate progress in a variety of areas. The techniques used are not new nor innovative by current standards. What is perhaps unique is the manner in which they have been applied to a large, complex, existing system.

The enhancements have been pursued in a systematic fashion. First, the software was extended to utilize the capabilities of the 2B Processor while still remaining functionally compatible with the original No. 2 ESS software. Commercial versions of these compatible programs were released in 1976 and 1977. Eventually the compatibility restriction was lifted and the software was extended to exploit noncompatible 2B capabilities. In about the same time frame, modifications to enhance functional and data independence were added.

3.1 2/2B compatible extensions

Functional compatibility between the No. 2 ESS and the No. 2B ESS at the time of conversion was important not only to minimize risk, but also to ensure that future enhancements applied to both systems. The types of enhancements that could be introduced were severely limited by this restriction. Generally, they took the form of increasing the maximum quantity of a particular resource that the system could handle.

A prime example is the memory spectrum. Immediately following the conversion, the 2B Processor was limited to the 256K-word spectrum of the No. 2 Processor simply because the software was tailored to handling 18-bit addresses. The program modifications required to handle 20-bit data addresses and exercise the full megaword spectrum of the 2B Processor were scattered throughout the software. A pervasive change of that nature with its attendant risk is precisely the kind of complication that was intentionally avoided during the initial conversion process. With the converted system functioning well, exploiting the million-word address spectrum could be attacked as a separate problem. It was scheduled, designed, and implemented in 1976.

The implementation modified all affected programs to be able to handle 20-bit addresses. Since both the 2 and 2B versions were modified, they remained compatible and both were able to handle the larger addresses. The No. 2 version would, of course, never encounter an address that exceeded 256K.

Similar extensions followed in a well-defined and controlled manner. The maximum number of scanners, peripherals used to sense the states of circuits and/or lines, was increased from 12 to 31. The number of buffers utilized to control peripheral units was increased from 12 to 20. Each extension was scheduled and implemented separately, thereby avoiding the coincident introduction of complex changes.

3.2 Noncompatible extensions

The number and nature of compatible extensions are limited. To make more fundamental structural changes to the software system required exploiting aspects of the 2B Processor for which there is no No. 2 counterpart. As an example, one could not begin to use the superior 2B instruction set since the No. 2 Processor did not contain those instructions. In like manner, the larger No. 2B ESS memory could not be utilized for program instructions. In 1977, the decision was made to decouple the new and the old systems, thereby eliminating the compatibility constraint on the program system.

This decision opened new vistas for improving the structure of the software system. The first step was to simply eliminate methodology

and operations that existed only because of the No. 2 Processor. For example, the No. 2 Processor cannot support relocatable programs. The 2B software system has now been converted to be mostly relocatable. Certain instruction sequences can be coded more efficiently using the 2B instruction set. Some sequences, in fact, were not needed at all and were simply eliminated. We were particularly fortunate to have available the excellent text processing facilities provided by the *UNIX** time-sharing system to aid in the search for these sequences.[4] These changes were collectively called program "clean up" and were performed as the first step in moving primary system development to the No. 2B ESS.

3.3 Modernizing the control structure

3.3.1 Isolating application software from the control programs

One definition of an operating system is a set of standard functions that provide a hospitable environment for application programs. The call processing control mechanism in the No. 2B ESS did not supply this feature. Nevertheless, certain steps toward providing common, general functions for use by application programs were possible. Emphasis was on the introduction of these functions with a minimal perturbation to the overall system. In particular, designs that did not cause existing programs to become inoperative were selected for the implementation of these functions. Once a function is available, all new development can use it. On an independent schedule and subject to independent evaluation, a plan could be developed to convert existing application programs to use the new functions.

3.3.2 Isolating application software from a resource manager

This philosophy of gradual introduction has also been used to reduce the coupling between an existing resource manager and the application programs. The manager is responsible for finding and allocating paths through the switching network. To efficiently use the network, it attempts to share a portion of an existing path with the next path to be found for the same call. The shared portion is known as the A link and the operation is known as A-link sharing. As originally designed, the manager provided the mechanism, but the application programs were required to save the identity of any A link to be shared and to specifically request sharing when invoking the manager.

Consistent with this new philosophy, a universal mechanism for sharing A links has now been implemented. The path manager now has complete control over the sharing mechanism. Sharing is at-

---

* Trademark of Bell Laboratories.

tempted in every situation and not just at the discretion of the application program. The A-link sharing information has been eliminated from the application interface. The significant point is that once again this was accomplished in a fashion that did not require the wholesale change of existing programs. Unchanged application programs would still go through the motions of making A-link sharing decisions, but these would be ignored by the manager. In no way would the application program affect or compromise the control now centered in the path manager. All known A-link code was in fact stripped out of application programs for reasons of clarity and efficiency. Significantly, this was a conscious decision and not a foregone conclusion. In the event that some application A-link code was not detected, only the efficiency of the system and not its integrity was affected.

IV. MODERNIZING THE DATA STRUCTURE

4.1 Isolating application software from data

The original application interface to the No. 2 ESS database system (it was known as translations in those days) was reasonably well structured. All of the update routines were concentrated in one place, as were the access routines. Most application programs even used the access routines to retrieve the data. The fundamental problem was the transparency of the access routines. While they shielded the user from the gross structure of the data, returning the address of an individual record was not unusual. That practice, while promoting the ultimate in efficiency, resulted in the known record formats propagating throughout the application programs.

This data problem is more insidious than the corresponding control problem. It can be, and was, attacked with the same philosophy of installing the new without breaking the old. While exhibiting many of the same benefits as the new control interface, the database does not really pay dividends until the new interface is used to streamline the access and packing efficiency by allowing the records to be reformatted. All programs, including preexisting ones, had to be decoupled from the data by the new interface before any records containing that data could be modified.

Despite this observation on the general intractability of the problem, where it made sense to reorganize a portion of the database, a two-phased approach was used. The application programs were first decoupled from the data by a uniform and opaque interface supplied by the database programs. Since the underlying data structures did not change, converted, nonconverted, and "unaffected" data accesses continued to operate properly. The "unaffected" class contained some accesses for which the need for conversion was not predicted. The

existence of these oversights did not affect the operation of the system during the entire conversion process. Even assuming perfect analysis, the two-phased approach eliminated any coordination in the data access conversions. It was only necessary that all conversions occur before the database was modified.

The data were reformatted during the second phase. When the data changed, implicit and embedded data dependencies that were overlooked in the first phase surfaced and were dealt with. Certainly the disruption of the system was minimized by a two-phased approach. In the case of the No. 2B ESS conversion being reported herein, the entire decoupling of the application software from the data was not accomplished during Phase 1. Some converted code had to be reexamined and modified during Phase 2. Nevertheless, the final result was very successful based on the relative absence of conversion problems uncovered during testing. This method allowed the complete reformatting of the originating and terminating translators, which in turn allowed more features and provided a more regular data structure that could be enhanced gracefully.

4.2 Isolating application software from hardware

Superimposing a logical structure and nomenclature on top of the physical one is a well-established mechanism for insulating application software from the details of the real world. Application programs typically direct output, not to a real device, but to a logical channel number. The operating system is charged with mapping this logical number to a real device. Unfortunately, the need for similar flexibility in identifying terminals of the No. 2 ESS switching network was not foreseen. There was but one type of network whose terminals were universally identified by their actual equipment location. With the introduction of a second type of network, the use of equipment location numbers to identify terminals became ambiguous. Rather than imbuing all application programs with the knowledge of two network types, a logical identification scheme similar to the one used for channels was created. The logical scheme identifies each terminal by a unique virtual equipment number. A mapping algorithm is used to go from real to virtual equipment number and vice versa.

The merits of a virtual numbering scheme are self-evident and will not be discussed. The feasibility of introducing a virtual numbering scheme into a large, complex software system developed almost two decades ago for a modest development effort is less evident. The 2BE3 generic program is an existence proof.

First, a few sensible limitations were imposed. The spectrum of the virtual equipment numbers was made to coincide with that of the original real equipment numbers. This, of course, means that the

maximum number of actual terminals of all types is still the same. The required performance of the No. 2B ESS is such that this is a reasonable restriction. Making the spectrums compatible means that all existing programs and data structures could at least handle the identifying number. The change would, therefore, be entirely transparent to all programs that used the number strictly as an identifier. The new number could still be passed as a parameter, stored in the same slot in the call control block, and used to index into data tables to retrieve per terminal information. Only the programs that actually required the real equipment number (equipment control programs and programs that print equipment locations for human use) needed to be modified. The subset of affected programs was relatively small. In addition, the required modifications were reasonably straightforward. Functions to do the mappings were added to the database interface. The modifications to other programs consisted mainly of inserting calls to these new functions.

The virtual equipment numbering scheme is now in place and functioning well. Because its introduction is thought to have saved development effort in other areas of the project, the consensus of opinion is that its overall development effort was less than other considered methods. It will, of course, continue to provide benefits in the future when, and if, additional network types are added to the No. 2B ESS.

V. FEATURE EXTENSIONS

No. 2 ESS began on a small suburban switching machine. In the early 1970s, centrex features were added. With introduction of No. 2B ESS in 1976, real-time capacity was almost doubled, allowing even more business-type features.

The 2BE3 generic added compatibility with the 10A RSS (using the virtual terminal concept and the database modernization), EADAS, and AMACS, while allowing greater penetration of custom calling features (via the database modernization). Several of these features are described in accompanying articles.

VI. CONCLUDING REMARKS

The No. 2/2B ESS experience demonstrates that a mature system can evolve its hardware and software systems to more modern technologies. The successful methodology included making only one major change at a time, and, where possible, introducing a modern software structure without, or at least before, breaking the existing structure. The evolution reported herein spans the 1970s. The No. 2B system was proposed in 1971. It was developed during the middle years of the

decade and was placed in commercial service in February 1976. The compatible software enhancements occurred between that time and the next release of the system in December 1977. The noncompatible enhancements have occurred since that time and continue at present.

The software changes as a group have simplified software administration and the introduction of code changes to support new telephone features. The development effort has been quite modest. Introducing standardization has introduced some inefficiencies. Certainly the general-purpose control and data functions require more memory and in many cases a small real-time overhead. These penalties are not a large price to pay for the advantages of improved software structure.

VII. ACKNOWLEDGMENTS

Several people were instrumental in the No. 2B ESS concept and its original design. These included J. A. Herndon, C. E. Ishman, W. R. Nehrlich, P. D. Mandigo, and P. C. Richards. Beyond these few, the authors wish to thank the entire No. 2/2B ESS development organization that made the system a success. This paper is based in part on earlier works by Mandigo and Nehrlich.

REFERENCES

1. Special issue on No. 2 ESS, B.S.T.J., *48,* No. 8 (October 1969).
2. P. D. Mandigo, "No. 2B ESS: New Features From a More Efficient Processor," Bell Lab. Rec., *54,* No. 11 (December 1976), pp. 304–9.
3. T. F. Storey, "Design of a Microprocessor Control for a Processor in an Electronic Switching System," B.S.T.J., *55,* No. 2 (February 1976), pp. 183–232.
4. L. E. McMahon, L. L. Cherry, and R. Morris, "UNIX™ Time-Sharing System: Statistical Text Processing," B.S.T.J., *57,* No. 6, Part 2 (July–August 1978), pp. 2137–54.

AUTHORS

John E. Yates, S.B.E.E., 1960, S.M.E.E., 1962, Massachusetts Institute of Technology; Bell Laboratories, 1963—. Before coming to Bell Laboratories, Mr. Yates was a research assistant at the Massachusetts Institute of Technology. In 1962 he joined The Boeing Company. Since 1963, he has worked extensively on the No. 2 and No. 3 ESSs. Mr. Yates is currently responsible for development of application database change programs in No. 5 ESS.

AUTHORS

Thomas E. Grassman, B.S.E.E., University of Arizona, 1961; M.E.E., New York University, 1963; Bell Laboratories, 1961—. Since coming to Bell Laboratories, Mr. Grassman has worked on a variety of 101 ESS and No. 2/2B ESS assignments and on the establishment of the BX.25 standard for data-link protocol. Presently, he is Supervisor of a No. 2B ESS call processing software group.

THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 6, July–August 1983
Printed in U.S.A.

Modernization of the Suburban ESS:

Adding Data Links to an Existing ESS

By C. E. ISHMAN,* R. B. SANDERSON,* L. M. TAFF,* D. P. TRUAX,* and C. T. TULLOSS*

(Manuscript received June 21, 1982)

As a major modernizing improvement data links for three important functions have been added to the 2B Electronic Switching System (ESS). The data-link system was designed with a layered architecture using an enhanced subset of the X.25 protocol. Levels 1 and 2 of the protocol are implemented in the hardware and software of a new device, the Serial Peripheral Unit Controller/Data Link (SPUC/DL). Within the SPUC/DL itself, check circuitry and software are used for error detection. A second SPUC/DL for each critical application provides redundancy on either a dedicated or dial-up basis. An elaborate diagnostic program within the 2B ESS can be invoked automatically or manually. This paper discusses some of the issues in retrofitting the BX.25 protocol to pre-X.25-speaking machines.

I. INTRODUCTION

This paper describes the addition of real-time data-link capability to the No. 2B Electronic Switching System (ESS)†. The paper combines a technical description of the project with explanations of certain

* Bell Laboratories.

† Acronyms and abbreviations used in this paper are defined at the back of the Journal.

key decisions. We attempt to be candid in assessing both the virtues and pitfalls of such decisions.

The No. 2B ESS is a medium-sized telephone switching system designed for rural and suburban offices with 5000 to 20,000 customer telephone lines. The addition of data links to the system was but one of several major enhancements in the development of a new generic program developed for the 2B—the 2B Extended feature generic #3 (2BE3)—described in accompanying papers.

1.1 Project goals

Our major goal in this project was to create appropriate hardware/software systems to provide data-link capabilities from the 2B to each of three different remote Bell System *applications.* Each application had different characteristics and reliability requirements. An overview of the desired network capabilities is shown in Fig. 1. Note that other types of ESS machines as well as additional 2B ESS offices may interface to the Automatic Message Accounting Recording Center (AMARC) and Engineering and Administrative Data Acquisition System (EADAS) applications. This constrained the freedom to choose interface specifications freely. We wanted flexibility in our design so various combinations of these applications could be configured without trouble or extra cost.

A secondary goal was that as much as possible of the data-link subsystem resulting from our efforts be "portable;" the system should be adaptable to other applications on the 2B, and even other machines, after suitable hardware alteration.

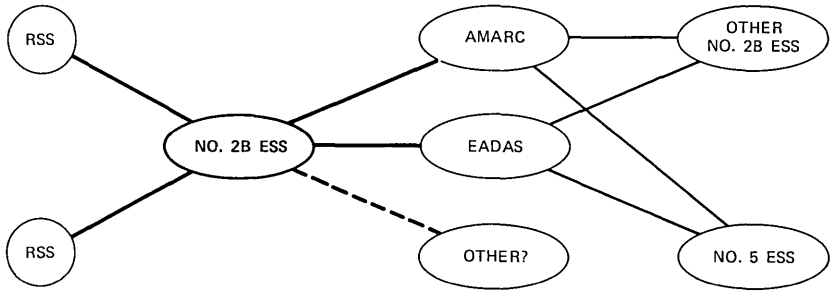
1.2 Boundary conditions

1.2.1 Use of BX.25 protocol

The development of our project proceeded roughly in parallel with the developments of the AMARC and EADAS projects within Bell Laboratories, as well as with the No. 5 ESS project, which was also to interface with AMARC and EADAS. A general consensus was reached to adopt recommendation X.25 of the *Comite Consultatif International Telephonique et Telegraphique (CCITT).* BX.25, an enhanced subset of X.25, has since become a standard within the Bell System. Its adoption had important consequences in the total design of a data communications system, as will become clear below.

1.2.2 10A Remote Switching System

The 10A Remote Switching System (RSS) is a small space-division system intended for rural communities with fewer than 2000 subscribers. It requires a larger *host* machine to switch interoffice calls. Its controlling program is resident in firmware, and has been developed



AMARC – AUTOMATIC MESSAGE ACCOUNTING RECORDING CENTER
EADAS – ENGINEERING AND ADMINISTRATIVE DATA ACQUISITION SYSTEM
RSS – REMOTE SWITCHING SYSTEM

Fig. 1—Global view of planned No. 2B ESS.

over several years in conjunction with the No. 1/1A ESS, which provided host capability. The 10A RSS feature of the 2BE3 generic program will allow the 2B ESS to be a host as well. Up to 31 RSS machines can be supported by one 2B ESS.

The 10A RSS, having been developed before BX.25, constrained the generality of the data-link system. Nearly 100 units were in service as the 2BE3 generic was being completed. We were required to interface to the RSS with no changes to its firmware.

II. GLOBAL SYSTEM ARCHITECTURE

We address in this section the architectural issues, both hardware and software, involved in the next level of detail from Fig. 1. For this purpose it may be useful for some readers to review briefly the structure of the BX.25 protocol.

2.1 Overview of BX.25

BX.25 is a *layered* protocol consisting of independent layers or *levels* (see Fig. 2). Each level at one side of a data link communicates (in a logical sense) with the corresponding level on the other side. We use three levels, not counting the application program. A call-processing program (say) believes it is conversing with a billing program, for example, at the AMARC. To send a *message* to its counterpart the program simply calls a protocol level 3 interface routine. For its part, the conversation is complete.

Level 3 software will break the message, if necessary, into *packets* of length appropriate to the implementation (depending for practical reasons on the kind of transmission facility, available buffering capacity, etc.). It will then send the packets to the level 3 software at our destination, expecting—as one of its standard functions—a "message-

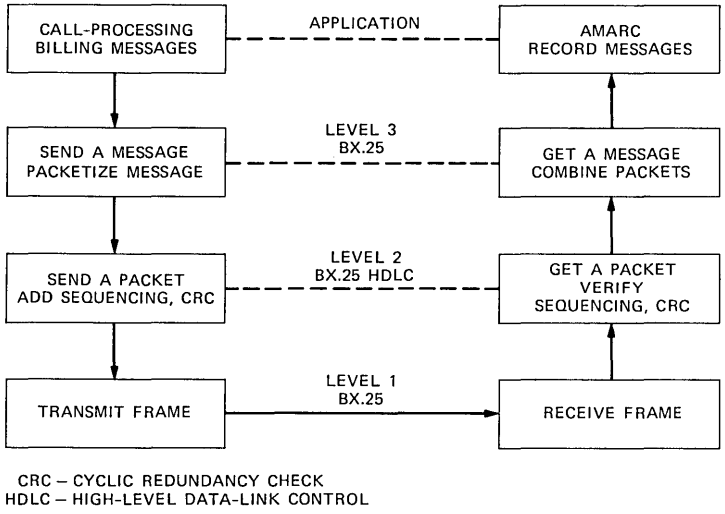


Fig. 2—Application layer and three layers of BX.25.

received" *acknowledgment.* It normally will not rest easy until such acknowledgment is forthcoming within a timed interval; it typically will retransmit the packets until a further interval expires, when it will begin recovery actions. Note, however, that "sending the packets" simply means handing them off to level 2 software.

Level 2 software is very analogous to level 3 in that suitable numbering is attached to the data, which are then delivered as *frames* to level 1.

Level 1 is implemented directly in hardware and is responsible for physically transmitting the data to the remote site. Level 2 frames are specified to contain a Cyclic Redundancy Check (CRC) at their trailing end, though in practice this is computed and inserted by an industry standard hardware chip. Thus there is a slight blurring of the line between levels 2 and 1.

While level 1 is supposed to transmit the data, the primary task of level 2 is to ensure that they arrive safely. Level 2 will retransmit unacknowledged frames, occasionally transmit *receiver-ready* messages on a temporarily unused link ("idle line assurance"), and handle initialization and disconnection functions. The numbering and acknowledgment functions of level 3 are primarily for flow control rather than verification purposes. A primary function of level 3 not yet mentioned is the routing of *logical channels* to their proper physical destinations.

Readers interested in more details about this protocol are referred to Refs. 1 through 3.

2.2 Architectures considered

One possible configuration would be to attach data-transmission hardware directly to the 2B ESS and write level 3 and level 2 software for the ESS. All the detail work of keeping track of unacknowledged frames, and time-consuming interrupts from transmission and reception, would be incurred by the switching machine itself. Since such overhead increases with the number of data links, real-time constraints made this an unrealistic option. In particular, we note that the link traffic increases just at least opportune times—when the machine is busy servicing telephone calls and has the least time available to handle link work. This configuration was never considered seriously. All realistic options use a separate autonomous processor to off-load the ESS.

We will discuss three such options given serious consideration. In the first option, level 3 and level 2 software are wedded together within a small auxiliary processor also containing the data-link hardware. In the second option, level 3 software resides with level 2 software capable of handling multiple data links, many of which are installed on the separate processor. The third option splits levels 3 and 2 into the ESS and attached processor, respectively, with level 2 servicing only one data link. This last option was eventually adopted.

2.2.1 Monogamous levels 3 and 2 together

A configuration with levels 3 and 2 residing in a microprocessor was chosen by the designers of the AMARC side of our data network for their "AMARC Protocol Converter" (APC). This choice has the advantage that level 3 as well as level 2 is off-loaded from the main processor. The disadvantage of concern to us is that when it is necessary to change traffic to the standby link in a multi-link configuration, all level 3 current parameters as well as *all data currently queued at level 3* must be unloaded from one device and transferred to another. At the AMARC side of the link this is not as serious a matter as on the ESS side, where large volumes of data may be generated.

2.2.2 Polygamous level 3 with multiple level 2's

A data link device for up to 16 independent links controlled by common level 3 and level 2 software is currently in use with the Bell System No. 1/1A ESS. This device, named the Peripheral Unit Controller (PUC), was the original model for our own system. Such a configuration has the advantage that links can be changed without intervention of the main processor. However, the PUC turned out not to be suitable for a machine the size of the 2B ESS; with enough power to drive 16 links there is a mismatch in the capacities and costs of the 2B ESS and the data-link controller. This consideration changed the

original direction of our project away from the PUC. Our device was originally to have been a PUC modified from parallel communication with the host to high-speed serial (6.67 Mhz); hence the name Serial Peripheral Unit Controller/Data Link, or SPUC/DL.

2.2.3 Levels 3 and 2 living separately

To answer the previous objection to high overhead cost for the office with few data links, we wished a small and inexpensive basic hardware unit. Vulnerability to failures could then be countered by duplicating units on applications requiring high reliability. An individual link need only be enough more reliable than the communications channel not to add significantly to the failure rate. Furthermore, difficulties in changing links could be reduced by keeping level 3 of the protocol and level 3 buffers within the ESS, which is fully duplicated, including all writable memory. In addition, where a controller implementing Level 3 would almost certainly need to be duplicated and have elaborate checking and matching circuitry to meet reliability requirements, the controller implementing Level 2 can use a simpler reliability check such as parity. Finally, of the three configurations discussed, this last seemed to us to be the most flexible, as well as the most fully portable to other applications and/or hosts. With this scheme substantial new code needed to be written for the 2B ESS—the level 3 protocol, and the maintenance software for the links. Furthermore, complete off-loading of the ESS processor does not occur. Though most of the work and interrupt processing is done in the SPUC/DL, there will be a small traffic-dependent load for managing level 3.

2.3 Configurations by application

As we mentioned previously, each of the three initial applications for the SPUC/DL had its own requirements and the simple hardware could be configured separately for each.

2.3.1 10A RSS

Data links for each RSS are fully duplicated with hard-wired dedicated lines for both links, thus providing a "hot" spare. It is recommended that diverse transmission facilities be used for each data link. These rather strong requirements underscore the high reliability expected of this application. Call processing for lines connected to the 10A RSS is done on the host machine, in our case the 2B ESS. The data links are used to control the progress of calls. Thus, loss of data-link capability means loss of most service for RSS customers. (The RSS can be configured to provide local calls if the host connection becomes lost.) The 10A RSS machine is described at length in Ref. 4.

2.3.2 AMARC

Traffic on the AMARC data links does not carry information that affects customer service, and therefore does not have the intrinsically stringent reliability requirements imposed on RSS links. However, the billing information carried is valuable, and long outages could be costly. Short outages—some tens of seconds—would be tolerable if suitable buffering capacity were provided to avoid losing billing data. AMARC links were therefore configured with one data link on a dedicated hard-wired line, supplemented with a "cold" spare connected to the switched network. Changing to the cold spare after a dedicated primary link outage exceeding some minimum duration involves dialed telephone calls between the ESS and AMARC machines. Such a scheme saves the considerable cost of a second dedicated connection.

2.3.3 EADAS

Though data gathered on EADAS links are useful for planning telephone network growth, their temporary loss does not compromise either customer service or revenue. EADAS applications can therefore be configured with a single data link, equipped with a dedicated line.

2.4 Architectural summary

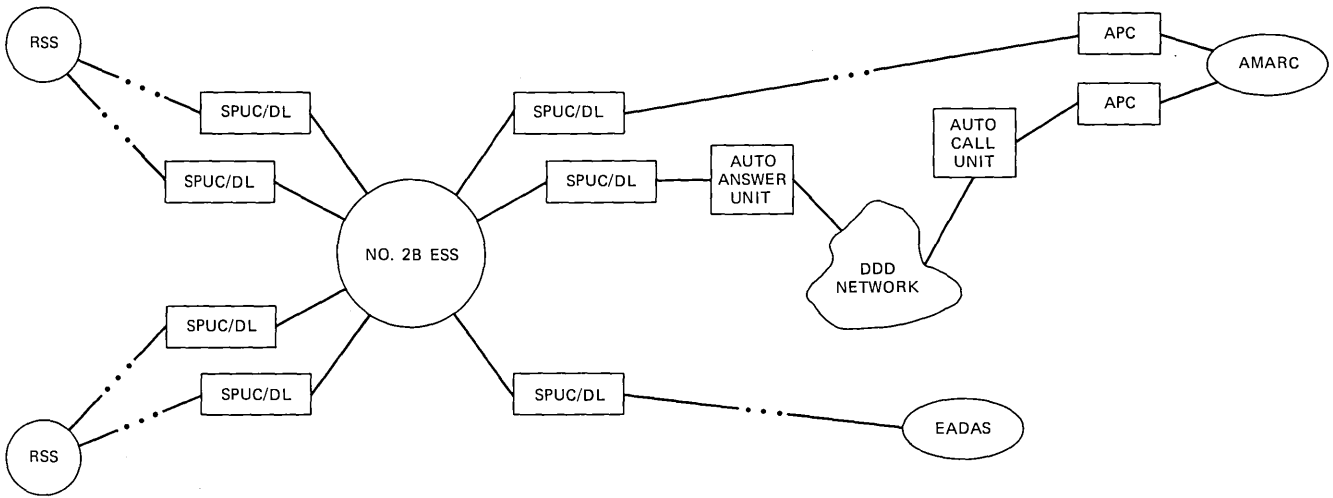
A diagram of a No. 2B ESS with its three types of configurations is shown in Fig. 3. We emphasize that the duplication on critical applications is for reliability; only one link is active at any time, and there is no load sharing.

III. FUNCTIONAL REQUIREMENTS OF THE SYSTEM COMPONENTS

3.1 Hardware

The general requirements of the hardware are as follows:

1. Basic hardware. Hardware for the data-link controller must execute level 1 (the physical level) of BX.25, and provide enough memory space and processing power to implement level 2.
2. Link speeds. Initial applications were to use 2400 or 4800 bits/second (bps), but for possible new applications a goal was set for 56K-bps hardware capability.
3. Link Reliability. Our goal was to have data communication failures be caused predominantly by transmission facility failures rather than by data-link controller failures.
4. Repairability. The controller required self-test capability to isolate faults to a single-printed circuit board wherever possible.
5. Cost. The initial cost of the data links was required to be low for smaller offices with only a few links.
6. Expansion. Expansion should be modular, smooth, and trouble-free.



AMARC – AUTOMATIC MESSAGE ACCOUNTING RECORDING CENTER
APC – AMARC PROTOCOL CONVERTER
DDD – DIRECT DISTANCE DIALING
EADAS – ENGINEERING AND ADMINISTRATIVE DATA ACQUISITION SYSTEM
RSS – REMOTE SWITCHING SYSTEM
SPUC/DL – SERIAL PERIPHERAL UNIT CONTROLLER/DATA LINK

Fig. 3—The three data-link applications of the 2B ESS.

3.2 Software for the SPUC/DL

Stated simply, the SPUC/DL software requirement was to implement level 2 of the data-link protocol. Unfortunately, the three data-link applications for 2BE3 required two sets of protocol rules for level 2. The AMARC and EADAS applications use the procedures defined by the BX.25 specifications,[3] which were derived from the CCITT X.25 "Link Access Procedure B" or LAPB. The third application, RSS, uses the older version of the X.25 level 2 procedure "LAP." LAP and LAPB, though similar, are not compatible (LAP cannot communicate with LAPB). However, except for the initial handshake, some error recovery procedures, and a few differences in command formats, the core of information processing procedures is the same.

Initial SPUC/DL applications were to use link speeds of 2400 bps, although future applications of AMARC requiring 4800 bps were foreseen. These requirements can be easily met by a microprocessor device.

Data traveling over common carrier facilities are inherently susceptible to any noise disturbances afflicting such facilities. It is essential that data altered during transmission be detected and the correct data retransmitted. This is the *raison d'etre* for level 2 of X.25. To this end, X.25 employs a 16-bit Cyclic Redundancy Check (CRC) to detect frames altered by noise, and a sequence numbering scheme to facilitate retransmission of frames discarded because of bad CRC. Flow control procedures at level 2 also prevent data loss from buffer overflow.

An additional requirement for maintenance purposes was for some diagnostic capability "on board" the SPUC/DL. The on-board (resides in SPUC/DL memory) diagnostic was to be used by the ESS diagnostic to test hardware inaccessible to the ESS or requiring excessive ESS resources to test. When the ESS wished the SPUC/DL to begin its diagnostic routines, it was to instruct the SPUC/DL operating system to relinquish control to the on-board diagnostic programs. To go back to normal the ESS must issue a master reset of the SPUC/DL, since some diagnostic procedures would destroy the normal operating state.

Diagnostic requirements include a "loop around" test in which data packets would be sent out on the link, directly echoed back by the remote end, and received by the SPUC/DL and 2B ESS. This would normally be a protocol violation in the use of an *address* field at the frame level. We have enhanced the implementation to suspend normal protocol address checking when the diagnostic is running.

3.3 Software within the 2B ESS

We discuss here our requirements for both the protocol software and the maintenance, or fault-tolerant, software. Most of the protocol

requirements are general and serve to elaborate on our previous discussion of BX.25.

3.3.1 Protocol-implementing software

The 2BE3 generic program requires dedicated point-to-point (permanent virtual circuits), full-duplex, 2400 (or 4800) bits per second communication facilities. The No. 2B ESS software implements only those portions of the BX.25 specification needed to meet the 2BE3 generic requirements.

Level 3 of the BX.25 protocol controls the transfer of data between level 2 and the next higher level (the application layer in No. 2B ESS). Transferring includes several processes, namely: packetization, multiplexing, sequencing, and flow control.

3.3.1.1 Packetization. Packetization is the process of a transmitter's dividing large messages into practical-sized packets, transmitting the packets, and having the receiver concatenate the packets to form the original messages. BX.25 does this by setting a "more-data" (M) flag bit in the level 3 header in all but the last packet of a message. The AMARC billing data messages have a maximum of 512 bytes; EADAS thirty-minute traffic data messages vary, with an average of 4,000 bytes depending on the central office. The packet size of 256 bytes was determined appropriate for the throughput desired and the expected error rate of the data transmission facility. If the packet size is large and the transmission facility error rate is high, many packets may receive errors and need to be retransmitted, which decreases the throughput.

3.3.1.2 Multiplexing. Multiplexing is the process whereby a transmitter combines data packets with identical destinations but associated with various application functions (logical channels). The receiver separates (demultiplexes) the packets via the logical channel numbers and distributes them to the appropriate application function. AMARC has a billing-data logical channel, EADAS a traffic-data logical channel, and both applications have a time-of-day logical channel; thus, multiplexing is required with both the AMARC and EADAS applications.

3.3.1.3 Sequencing. Sequencing of BX.25 level 3 data packets is required to assure packets are received in the identical sequential order of transmission. Each level 3 data packet header contains two modulo eight sequence numbers: a send sequence number P(S), which identifies the data packet itself, and a receive sequence number P(R), the expected number of the next data packet received.

BX.25 permits a level 3 transmitter to send a prescribed number of packets to level 2 without obtaining acknowledgment that earlier

packets were received correctly. These unacknowledged level 3 packets sent to level 2 are in the *window* until their reception has been acknowledged. The *window size* of each logical channel is the number of packets that can be sent to level 2 without being acknowledged; the AMARC window sizes are both five while EADAS windows are both two.

3.3.1.4 Flow control. Flow control is provided by BX.25 to permit receivers a method to regulate the rate at which data packets are sent to it. A packet received with send sequence number n is acknowledged with the next packet sent by setting the receive sequence number to $n + 1$. Any receiver having a data-capacity problem may delay the acknowledgment of received packets and avoid data loss.

3.3.1.5 Requirements for the 10A RSS. RSS level 3 was originated with No. 1 ESS before the acceptance of X.25 and is not compatible with BX.25. RSS level 3 messages contain a unique "SYNC word" in the first two bytes. Received messages are discarded if they do not begin with a SYNC word. The SYNC word is manipulated entirely by the RSS level 3 program and is transparent to the No. 2B ESS application call-processing software.

A two-byte remote terminal header following each SYNC word contains the RSS message type in a five-bit "client identity" field. The level 3 software distributes (demultiplexes) received messages to the call-processing application programs by interpreting the client identity. An RSS level 3 header also contains an eight-bit word count field specifying the number of data words in the message.

Packetization of messages is also used in RSS level 3 communication. The maximum RSS packet size is sixteen bytes and the packet boundary is totally asynchronous with messages from an RSS. RSS-bound messages transmitted from the No. 2B ESS always begin a new packet but may require several packets, depending on their length.

Communication concerning only the No. 2B ESS and SPUC/DLs is passed in *control packets* with a unique packet descriptor. A *packet descriptor* is the first two bytes of *all* packets passed between the 2B ESS and a SPUC/DL, identifying the packet type (data or control) and specifying the packet length in bytes. Packet descriptors are not transmitted over the data links. Control packets are external to level 3 and pass directly between the SPUC/DL and its maintenance configuration and diagnostic programs.

3.3.1.6 Timing requirements. We have mentioned previously that the decision to put level 3 in the 2B ESS imposes a penalty in traffic-dependent processing power. In requirements terms, level 3 had to be fast.

3.3.1.7 Other requirements. It was foreseen that part of the protocol-handling software package would include low-level routines for physical I/O to each SPUC/DL. Two requirements were imposed on these I/O routines: error routing and monitoring. Packets whose reception was not completely normal were to be routed to the data-link maintenance software. This includes nondata (control) packets, I/O errors, and packets from a nonactive SPUC/DL. A manually invoked monitoring utility was to report part or all the data sent to and read from an application. The monitor proved to be of value during system debugging and integration.

3.3.2 Maintenance (fault-tolerant) software

There were several general requirements assigned to the data-link maintenance function.

1. Initialization and Loading. On certain rare events there may be occasion to reinitialize any or all links in the system. This may require a down-load of the SPUC/DL program from its magnetic-tape residence at the ESS. This would also be done when field-updating SPUC/DL software.

2. Handling Error Reports. Maintenance software is required to maintain records of such reports, which may include frequent loss of facility carrier, low efficiency of transmission, etc.

3. Error Recovery/Link Reconfiguration. Serious faults (e.g., loss of communication with a SPUC/DL) or slow degeneration of a link may require removal of the affected link from service and changing of traffic to the standby link, if available. Thus, data-link maintenance is charged with trying to reconfigure a failing system to maintain traffic.

4. Regulation of the Diagnostic Program. The diagnostic, controlled by the maintenance software, is to be executed automatically when a failure occurs, and routinely on a nightly basis to detect faults on standby links before they are needed in service.

5. Audits. Maintenance "audit" programs were required to check periodically the legitimacy of the control-block database of data-link maintenance, to correct and report single errors, and to reinitialize if multiple errors were found.

6. Console Operator Interface. The final requirement of the data-link maintenance software was to interface with the human operator of the system, providing keyboard input service and output of various responses, as well as spontaneous reports of normal and abnormal events.

3.3.3 The diagnostic program

The diagnostic program had to detect and locate faults in the 2B

processor I/O interface, the SPUC/DL, and the component parts of the transmission facility. In the RSS application, it must also locate faults in the remote data-link control hardware. The diagnostic control structure was required to be sufficiently flexible to permit systematic testing during the installation of the SPUC/DL, the near- and far-end data facilities, and (for RSS) the remote data-link equipment. It had to be able to stress the overall data link enough to recreate faults observable only during heavy traffic.

IV. DESIGN AND IMPLEMENTATION—SPUC/DL HARDWARE AND SOFTWARE

4.1 SPUC/DL hardware

We can identify three key decisions in the evolution of our design. All three involved hardware/software trade-offs, the software being in the 2B ESS. The first decision involved the scope of the SPUC/DL, as we have discussed in Section II. The second decision was the use of periodic polling of the SPUC/DL rather than program-interrupt hardware. We will elaborate on this in Section V. The third decision was to use writable Random Access Memory (RAM) instead of Read-Only Memory (ROM).

4.1.1 Use of writable memory

The advantage of RAM is that updates of SPUC/DL software for circuits in the field are much less costly, requiring only software procedures rather than replacement of physical chips. The main disadvantage is that the memory is volatile; the program is destroyed (and must be reloaded by the 2B ESS) if power to the SPUC/DL fails; therefore, software within the 2B ESS had to be developed for downloading. In addition, the SPUC/DL software must have a nonvolatile residence from which it can be loaded. In our case this was to be the magnetic tape cartridge of the 2B ESS. The need for loading would cause concern only if long delays were incurred by the loading process. However, since 2B ESS-SPUC/DL communication proceeds nearly at memory speeds, the loading time mostly comprises the time it takes to retrieve the program from tape.

4.1.2 Logical design

The data-link controller itself consists of a single 16-bit microprocessor, its associated RAM, and two Direct-Memory Access (DMA) controllers, DMA1 and DMA2. Duplication within a single SPUC/DL is not necessary because of the relaxed reliability the distributed architecture allows. DMA controller 1 is a multiple-channel device allowing data transfers to take place between the 2B ESS and the

SPUC/DL independently of the microprocessor. Similarly, DMA controller 2 provides independent transfers between the SPUC/DL and the transmission facility (see Fig. 4). This architecture minimizes demands on the microprocessor while allowing high throughput. Level 1 protocol is incorporated in a Universal Synchronous-Asynchronous Receiver-Transmitter (USART) and standard interface chips.

4.1.3 Physical design

To meet the requirements of modular growth and high multiple-

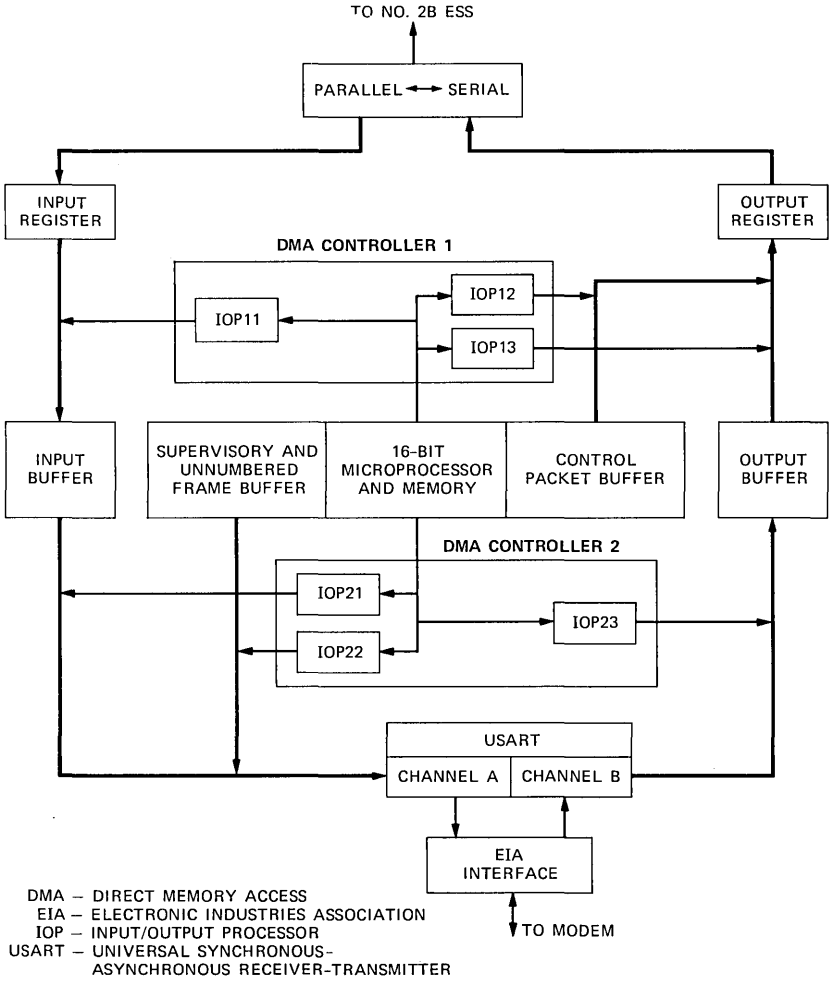


Fig. 4—SPUC/DL hardware components and architecture.

link reliability, each data-link controller was made a separate circuit. Pairs of controllers share a −48 volt power lead. Each SPUC/DL has its own power supply, transmission facility interface, microprocessor, memory, and communications channel to the 2B ESS. There are four printed circuit boards per SPUC/DL, containing the microprocessor and associated logic, memory, the data communications hardware, and the logic for I/O to the 2B ESS. Four complete SPUC/DL circuits, their associated power supplies, fuses, and connectors are housed in a single chassis.

A concentrated effort was made to provide a straightforward maintenance procedure. Each circuit has its own power fuse and modem cable. Each circuit is numbered, and this numbering is used on the circuit board enclosure, power supply housing, power fuse, and modem connector. In addition, each circuit is color coded, and the identifying color appears everywhere the circuit number is used. The 2B ESS SPUC/DL diagnostic program prints both the circuit name and color when it refers to a particular SPUC/DL.

4.1.4 Reliability issues

To provide sufficient reliability, the data-link controller's memory (program and data) is stored with parity and the parity is checked whenever data are read out or written. In addition, write-protection hardware is provided for the program memory. Any attempt to write into program memory will be prevented, and the 2B ESS will be informed. The status of the SPUC/DL hardware is summarized in a hardware status register that the 2B ESS can read.

The SPUC/DL on-board diagnostic program first computes a check sum over the program memory, tests the data memory, and then confirms proper operation of the microprocessor itself. Once the basic sanity of the microprocessor and correctness of its program is established, the diagnostic program goes on to test both the DMA controller that oversees data-link communication and the USART. The USART is tested by actually transmitting on the link. Additional loop-around circuits at the EIA interface, the associated modem, and the far-end modem allow faults to be isolated and identified easily. This philosophy of testing sections of the controller nearest to the central processor and then relying on those sections to test sections further removed allows economical and rapid fault isolation. Fault isolation is enhanced by partitioning the SPUC/DL into its four circuit boards. All interface circuitry to the 2B ESS is on one board. The second board contains DMA controller 1 and address decoding circuitry. The third board contains all data and program memory. The fourth board has the microprocessor itself, DMA controller 2, and the USART and modem

interface. Since the proper functioning of each board relies on the previous board but not on subsequent boards, most faults can be isolated to a single board.

4.2 Software within the SPUC/DL

Because of the commonality between LAP and LAPB, it was decided to implement both of them in one software load; the ESS would specify one or the other with a parameter sent to the SPUC/DL on initialization. Though this decision increased the space used for code in the SPUC/DL, it eliminated keeping track of which code to download into which SPUC/DL and the need for extra space on the cartridge tape. Separate codes would also have caused complications with the diagnostic. Perhaps the largest factor in the single-program decision was the expense of having to design, code, test, and administer two separate programs.

First we look at the SPUC/DL software structure from a functional data flow viewpoint, and then at the overall structure of the operating system that implements these functions.

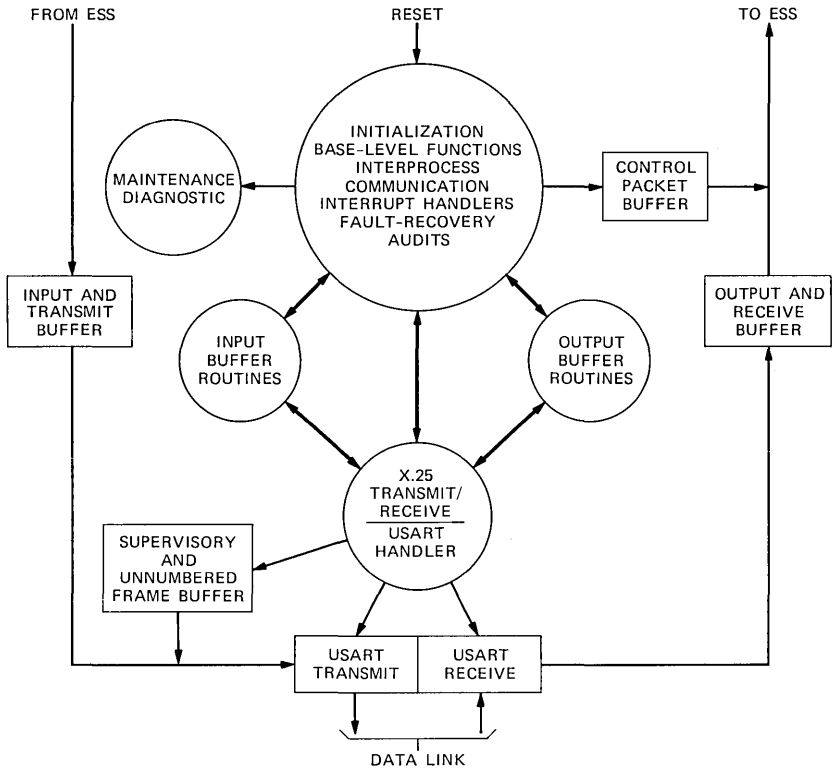
4.2.1 Data flow within the SPUC/DL

The hardware architecture of the SPUC/DL shown in Fig. 4 closely parallels the flow of data through the device. Internal to the SPUC/DL the send and receive processes run simultaneously. Data packets from the ESS enter the input buffer under control of DMA1 and are cataloged by the input routines, with *control* packets (messages for the SPUC/DL itself rather than for transmission) being processed immediately. Once packets have been cataloged, they are processed by the X.25 routines, which assign sequence numbers and set up the DMA2 transfer to the USART for output over the link as X.25 frames. Simultaneously with transmission, frames are coming in from the receive channel of the USART, are processed by the X.25 routines, and pointers to information frames are passed to the output routines. The output routines add packet descriptors to the information frames, and then set up the DMA transfer to the ESS.

This is a high-level view of the SPUC/DL and there are many other software processes involved in its operation, including maintenance routines, which play a large role in support of data handling.

4.2.2 SPUC/DL Operating System

The SPUC/DL Operating System may be visualized as a hierarchical multiprocessor operating system. In this system the 16-bit main processor acts as a master while the six individually programmable DMA channels are slave processors. Once a DMA channel has been



ESS – ELECTRONIC SWITCHING SYSTEM
USART – UNIVERSAL SYNCHRONOUS-ASYNCHRONOUS RECEIVER-TRANSMITTER

Fig. 5—SPUC/DL software processes; interprocess communication has been emphasized.

given a process, it will run that process to completion without further attention from the master. We denote these single channels as I/O processors, or IOPs. Most interprocessor communication is one way between the master and the IOP. Interprocess communication is handled using producer/consumer semaphores.

Figure 5 shows how the SPUC/DL software processes use the hardware to carry out the data communications function. Each I/O processor handles a specific I/O function. Communication between the SPUC/DL and ESS is handled by three channels of DMA1. In Fig. 4 they are labeled IOP11, IOP12, and IOP13. IOP11 runs the process of moving data from the input register to the input buffer. Once the SPUC/DL has been initialized by a master reset, the master processor starts this process running on IOP11. This process runs continuously and becomes inactive only when input buffer space is

exhausted or the process may be preempted by the microprocessor for status updates. This status information consists of "where will the next word be placed in the input buffer," or "deallocate *n* input buffer words." Unlike IOP11, IOP12 and IOP13 run only when there are processes scheduled for them (packets to be sent to ESS). IOP12 transfers control packets from scratch memory to the output register and IOP13 transfers data packets from the output buffer to the output register.

The DMA2 processors have almost mirror-image functions. IOP21 moves packets from the input buffer to the USART where they are serialized and sent out over the link. Packets are scheduled for IOP21 on a first-in/first-out (FIFO) basis with priority for packets being retransmitted. IOP22 moves supervisory and unnumbered frames from scratch memory to the USART. IOP23, which moves data in from the link, does not run continuously because of the intervention of the master processor to handle alignment of the start of incoming data. All other processes are run by the master processor, including the considerable primary task of processing X.25 frames.

4.2.3 SPUC/DL software reliability

In developing SPUC/DL programs considerable attention was given to reliable software design through modularity. Both the SPUC/DL Operating System and the level 2 procedures were coded in the C language and a top-down design structure was used wherever practical. Assembler-level code was used only where necessary for speed or to handle hardware-dependent functions. The high-level language had advantages both from an implementation and a testing standpoint. However, because both the hardware and software were new to us, during debugging it was occasionally necessary to look at the assembly code or binary output of the C compiler to determine the source of some problems.

The hardware of the SPUC/DL provided three features that also enhanced software reliability. First, the use of DMA hardware controllers for I/O eliminated the need for complex scheduling algorithms to move blocks of data. This is especially important in an I/O-intensive application such as a data-link controller. Second, write-protection hardware was provided. Since the program store of the SPUC/DL is in writable memory, without the write-protection feature of the hardware, the program would be vulnerable to bugs that alter the program itself. If a write-protect violation occurs, the SPUC/DL software is not altered but an interrupt routine attempts to save as much pertinent information as possible and then proceeds normally. The ESS is notified by a message from the SPUC/DL containing the hardware status register. The third hardware feature is a "sanity" timer, which

expires unless periodically serviced by the software. The timer guarantees that the microprocessor will not get stuck in any one place for an inordinate amount of time. If the sanity timer should expire, an interrupt is generated and is handled similarly to the write-protect error.

In addition to the hardware mechanisms mentioned, reliability is increased by periodic software audits. The audits check major global data structures, looking for inconsistent or out-of-range values. No attempt is made to correct audit-detected errors. The event is reported to the ESS, which prints a message.

4.3 System performance

The SPUC/DL software has an average one-way delay of about 4.5 ms. It has been tested at 5 to 10 times the speed expected in initial field use.

V. 2B ESS HOST SOFTWARE

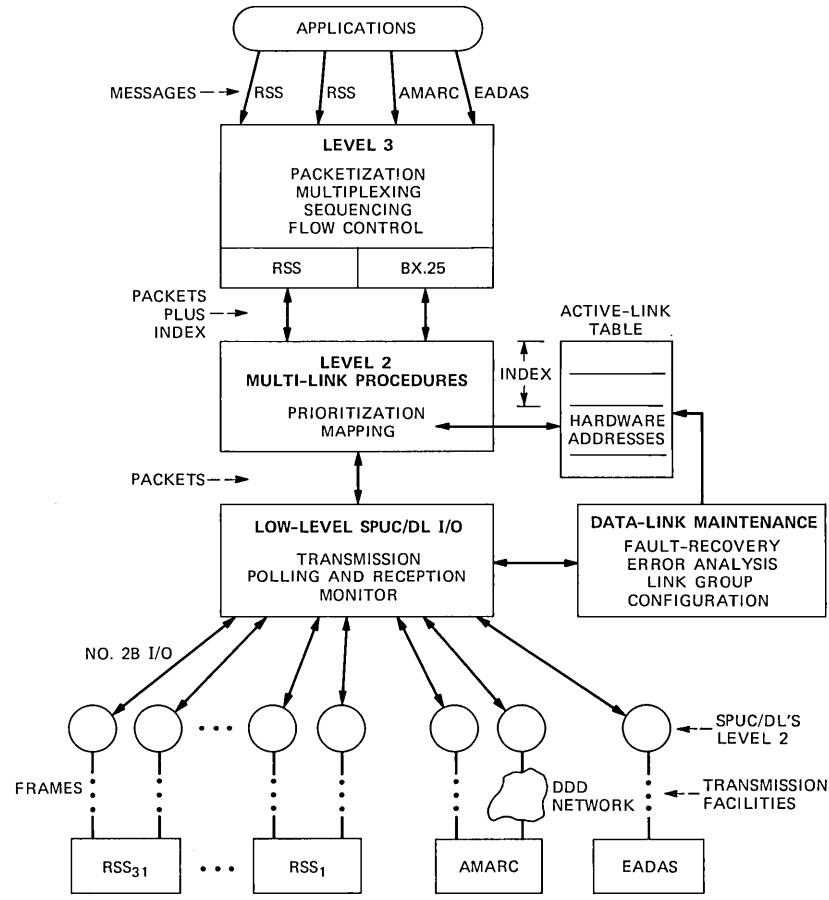
5.1 Architectural overview

Figure 6 is a high-level diagram of the data-link software in the 2B ESS. When they have a message to transmit, call-processing routines, billing routines, and data-collection routines enter appropriate points in level 3. Level 3 converts the messages into packets, breaking large messages into multiple pieces if necessary. Multiple logical channels (logical destinations) are sorted for their physical destination (multiplexing). Appropriate sequence numbers are attached. The packets are transmitted immediately, if possible, or put on a queue if flow is currently under control.

When the packets are ready to be transmitted, the software passes them to the level 2 Multi-Link-Procedures. These routines sort the packets by priority, and then do a *mapping* to a definite hardware address (i.e., physical SPUC/DL). The mapping is done through a table managed by the data-link maintenance software. Thus, the routines doing transmission need have no knowledge of which link they are using.

The Multi-Link-Procedures pass the packets to the low-level transmission software, which physically outputs them to the SPUC/DL whose hardware address was passed as an argument. The link-traffic-monitoring utility is implemented at this level. Level 2 in the SPUC/DL ensures error-free transmission to the destination.

The inverse process, packet reception and assembly into messages, is similar. Note that on reception, not all packets get routed to the Multi-Link-Procedures. Control packets originating within the SPUC/DL, and information packets from links not currently active are given to the data-link maintenance software. The maintenance



AMARC – AUTOMATIC MESSAGE ACCOUNTING RECORDING CENTER
DDD – DIRECT DISTANCE DIALING
EADAS – ENGINEERING AND ADMINISTRATIVE DATA ACQUISITION SYSTEM
RSS – REMOTE SWITCHING SYSTEM
SPUC/DL – SERIAL PERIPHERAL UNIT CONTROLLER/DATA LINK

Fig. 6—Data-link software within the No. 2B ESS.

software also uses the low-level common I/O routines to transact its business with each SPUC/DL. In this way the monitoring utility will report *all* communication to any link or link group.

5.2 Device polling and timing considerations

To have the SPUC/DL request attention from the 2B ESS, a polling method was adopted owing to complications in the 2B ESS demand-interrupt hardware. When it requires attention from the 2B ESS, the SPUC/DL sets a bit in its status register. The status register's bit controls a *ferrod*, a ferrite device whose magnetic state may be sensed

under program control by a *scanner*. These particular ferrods are arranged in rows of 16, so the status of 16 SPUC/DL "packet-present" ferrods may be read simultaneously. Once every base-level loop (100 milliseconds) the 2B ESS will poll all scanner rows containing SPUC/DL packet-present ferrods.

Clearly, such polling is inefficient; for polls on which no SPUC/DL has data, the polling time is wasted. The total time available for processing calls is reduced, and so therefore is the performance of the 2B ESS. The actual time penalty depends on how fast the software is. In our case the loss is around 0.6 percent with idle data links.

Of greater concern are the effects of measured-service billing (billing every telephone call) to the AMARC, and of RSS data-link traffic (several messages are required per call). In the AMARC case, the data links shorten the time spent on billing by about 30 percent relative to the previous use of magnetic tape. In the RSS case, only around 7 percent of the real-time of a typical call is due to the presence of the SPUC/DL. We can conclude that level 3 processing and programmed I/O to each SPUC/DL are not serious drains on the time available to the ESS, relative to its other tasks.

5.3 Maintenance software

The link(s) from the 2B ESS to any single remote site make up a *link group* of one or two links. It is the task of data-link maintenance to ensure that a link is always active in every link group.

5.3.1 Architecture

The highest-level component of data-link maintenance is called *link-group control*, and has responsibility for keeping a link always active in every group. Because of the substantial differences in the three main applications, link group control was divided into three independent sections, one handling the EADAS link group, one handling the AMARC link group, and one handling all RSS link groups.

Link-group control depends on a collection of utility routines called *link-state control*, which performs tasks on single links, independent of the type of link group of which the link is a member. Conceptually, a given link may be thought of as controlled by, or belonging to, either link-group control or link-state control (but not both) at any one moment. Link-group control manages links that are "in service," that is, actively carrying data ("active") or available for such use if needed ("standby"). Link-state control operates on links that are "out of service" by virtue of having been "removed"—either automatically after detection of a fault, or manually through issuance of a console keyboard command by the operator of the system.

The link-state-control routines depend in turn on a *loader* for down-

loading the SPUC/DL software from its residence on the system magnetic tape cartridge, and on the diagnostic program, which is described in more detail below.

The software of the 2B ESS continuously executes a fundamental "base-level loop" of some 100 milliseconds. Within the loop various functions are given control with the understanding that no function will exceed its own allotment of time. That is, instead of being driven by a clock interrupt, context switches (task changes) are the responsibility of the base-level routines themselves. They must save and restore their own contexts if completion of their functions requires more than one allotment of loop time. For this purpose and for keeping longer-term records, data-link maintenance makes use of control blocks.

5.3.2 Data structures

Two main types of control areas are maintained by the software. The link-group control keeps a link-group control block for each link group. In addition to current-function context information for link-group control, records are maintained of the number of changeovers between links in the group. This can prevent thrashing back and forth during periods of trouble, such as a lightning storm. In general, this block is intended for those records associated with the entire group, rather than with one specific link.

The second main type of control block is maintained by the link-state control routines, and is called a link-state control block. One such block is reserved for every link defined in the 2B ESS database. Here is stored information on the current context of link-state control for each link. In addition, a collection of records is maintained about the recent history of each link. For example, it is remembered at what time the diagnostic program was last run (to prevent too frequent use of the diagnostic), as well as its result. Over 30 fields are reserved for counting each of the types of errors reported by the SPUC/DL or recognized by the 2B ESS. Cumulative counts are also kept for each of several classes of errors, a class being defined by its relative seriousness.

The other major data structures used by data-link maintenance are the permanent (read-only) office records defining the data links present and their attributes (modem type, hardware address, etc.) These structures are accessed through special database routines.

5.3.3 Recovery strategies

The primary strategy for recovering from data-link failures is to use the redundancy engineered into the system—that is, to move link traffic to the other link in a two-link group.

The 2B ESS can detect failures of various natures and degrees of severity. The most obvious and straightforward is an inability to communicate with the SPUC/DL because of failures of hardware input or output orders. Here it is first attempted to change, if possible, to the standby 2B ESS processor to see if the orders can then be executed. If not, the fault is classified as *hard.* This is the fault detected when the power supply of a SPUC/DL fails (or is turned off).

A wider range of problems can be discovered by the checksum mechanism. During initialization sequences and when returning an out-of-service link to service, the ESS resets the SPUC/DL by setting a bit in the status register. This action forces the SPUC/DL program counter to an absolute address where its initialization code begins. As part of this code, a checksum is computed on the entire program memory of the SPUC/DL. The computed checksum is returned to ESS along with a prestored value it is supposed to match. The return of a matching checksum shows that the ESS can access the SPUC/DL, that the processor within the SPUC/DL is functioning, that the program is loaded and is probably correct, and that the device is in a known state. A bad checksum probably means a memory fault. In an emergency, a link can be "forced" into service even with a bad checksum comparison, so long as some checksum is returned.

Some symptoms are generally associated with the carrier facility rather than the data link. As mentioned previously, when troubles such as carrier loss are encountered, they are recorded. Noise on the facility will cause low link *efficiency*, defined as the ratio of bytes acknowledged properly to the total number transmitted. These troubles and a variety of others are individually counted within the data-link maintenance software. When the cumulative counts of such errors exceed some threshold value, recovery action begins. The counts of each type of event are weighted by their relative seriousness or impact on the system. The recovery strategy for these (individually) "nonfatal" errors is to compare the recent history of such errors for both links in an RSS link group, changing links if the standby SPUC/DL seems better. In an AMARC link group, since there is no guarantee that a standby link is available at the AMARC site, the 2B ESS must depend on the AMARC to detect facility problems and to change links. Only a severe problem will cause the 2B ESS to remove an AMARC link.

5.3.4 Changing links

The operation of changing traffic from one link to another in a link group is fundamental in the data-link recovery strategy, and required considerable development effort. The two applications for which the

operation is required, AMARC and RSS, are quite different. We will give a brief overview of each.

5.3.4.1 Changing AMARC links. The change operation on an AMARC link group attempts to follow the BX.25 protocol in most respects. Carrier is detected on the standby link after automatic dialing and answering hardware completes the connection via the DDD network. The ESS sees that the AMARC machine issued a "disconnect" on the primary link within its protocol level 2. Next, level 2 of the protocol becomes active on the standby link. Suitable security precautions are now taken to ensure the connection is genuine. Finally, level 3 of the protocol begins, and the operation is complete.

In the other direction the procedure is simpler. After the protocol has initialized at level 2, receipt of any information frame on the primary link will cause the 2B ESS simply to disconnect (hang up) the standby link.

5.3.4.2 Changing RSS links. For historical reasons, changing traffic between RSS links is complicated. The procedure was originally developed for the 1/A ESS. As we described in Section II, in the 1/1A environment both RSS links are connected to a single processor, the PUC/DL. Since there is no level 3 protocol for the RSS, resetting protocol level 2 (such as might be expected when changing links) has serious repercussions. Level 2 resets cause loss of data because they clear frame buffers. In the PUC/DL a changeover procedure was designed so no level 2 reset would occur and no data would be lost. The PUC/DL transfers the outstanding frames and current values of the protocol parameters (sequence numbering) to the new hardware by changing a pointer in its database. Thus, after changing links the old level 2 is running on the new link. Similar actions occur in the RSS; no reset of level 2 takes place, and no data are lost.

Since it was part of our design goal to leave RSS code unaltered, it was necessary for the 2B ESS and SPUC/DL to emulate the actions of the PUC/DL during the changing of the links. This was done by the following sequence: (1) protocol level 2 is brought up on the standby link, though transmission of data is disabled; (2) transmission stops on the active link while it is emptied of received data; (3) untransmitted and transmitted but unacknowledged data are read out of the active link and written into the standby link; (4) the protocol state of the active link is transferred to the standby to preserve the sequence numbering; (5) a "change links" message is transmitted to the RSS on the standby link, which turns the standby into the active link, and causes the RSS to change also.

5.4 Diagnostic software for the SPUC/DL

Fault location for the SPUC/DL data links can be best characterized

by its diversity. The diagnostic must contend with three different link configurations—AMARC, RSS, and EADAS—each having its own testing requirements. There are three execution environments: the 2B ESS, the SPUC/DL microprocessor, and for RSS, the remote terminal microprocessor. In addition, the diagnostic faces unusual conditions within the 2B ESS. It must be executable in the off-line processor, and must accommodate several instances of itself executing in parallel.

We have added two features to the diagnostic that are new to the 2B ESS. First, the SPUC/DL diagnostic specifies explicit replaceable units in the TTY output messages, rather than coded trouble location information. Second, to simplify installation and growth of a SPUC/DL configuration, we have implemented *partial* diagnostic capabilities—the ability, for example, to run only the first few tests so uninstalled hardware will not cause error messages.

To achieve flexibility in dealing with different link configurations, the diagnostic was organized for table-driven execution of independent tests, the table being specified either by the application type, as defined in the 2B ESS database, or by an operator-generated input message. This permitted maximum use of common diagnostic code, yet flexibility in tailoring specific tests for specific applications. The table also provided a simple means of providing special test functions not normally run—for example, an exhaustive SPUC/DL program memory test rather than the checksum test normally used by the routine diagnostic.

The diagnostic proceeds with the 2B ESS executing directly initial interface tests to the SPUC/DL, then requesting the SPUC/DL to run tests of its internal hardware. It then uses the SPUC/DL to do analog loop-back testing to the local modem, followed by digital loop-back to the remote modem, and finally (for RSS) requesting remote terminal hardware tests to be run. For those tests not executed by the 2B ESS, the diagnostic acts as a control program and analyzes data to convert the cryptic test failure information into human-readable messages specifying the replaceable unit.

While most of the diagnostic tests for the SPUC/DL data links use conventional diagnostic testing techniques, the digital (remote) loop attempts to find faults normally discovered under heavy-traffic conditions. This test sends 1000 sixteen-byte packets from the 2B ESS through the SPUC/DL, over the link, through the remote loop-back, and back to the ESS through the SPUC/DL. The SPUC/DL uses its operational link efficiency algorithms to quantify the quality of the link. A link is reported as a good link, a failing link, or a "degraded" link—usable but performing below what is considered optimum.

After a fault has been repaired, it is necessary to test the 2B-SPUC/DL interface from both the on-line and off-line central processors.

This ensures not only that faults in the off-line interface are located but that repair of an interface board has not introduced a fault in the standby processor to SPUC/DL interconnection.

The SPUC/DL diagnostic is the first program to take full advantage of a newly enhanced 2B ESS software subsystem for maintenance, which provides a degree of multitasking convenience. Use of the "Multi (base-level) Scan Function" (MSF) program allows for multiple simultaneous executions of the diagnostic on different data links with a single copy of the diagnostic and several instances of its data memory, one for each execution. The only significant complication is message routing from each SPUC/DL to the appropriate data memory, which is done through interfaces with the I/O and the data-link maintenance programs.

VI. PROJECT POSTMORTEM

6.1 Problem areas during development

Though the difficult problems of the project integration phase are freshest in our minds as this paper is written, there were problems of various kinds throughout the project. The most difficult *technical* problems in the data-link area were met in the integration and final debugging phase of the project. The problems were usually lost words of data, or protocol failures that stopped communication. They occurred randomly in time, usually when data-link traffic was high, and were not easy to reproduce. At least three factors interacted to increase the difficulty in solving the problems.

First, the SPUC/DL test facility did not have enough capacity to stress the links. Only the 2B ESS laboratories could drive the links at the acceptance-test levels, and time in these laboratories was in great demand by people working on all aspects of the 2BE3 generic program. The SPUC/DL test facility was improved, but only after it was clear that serious but infrequent problems were present in the SPUC/DL. The test-facility shortcomings meant that the SPUC/DL had more problems than it should have during the integration phase. The service-affecting problems had been mostly resolved by the time the test facility became really adequate.

The second factor lies in the distributed nature of our systems: with three asynchronous processors, it was often difficult to isolate where any particular problem arose—the 2B ESS, the SPUC/DL, or the application (AMARC, EADAS, or RSS).

Third, debugging tools for the system as a whole lacked some desirable features. The 2B ESS laboratories had powerful debugging hardware for halting the processor, dumping memory locations, tracing execution paths, etc. These facilities were not available on the SPUC/DL, particularly in the late stages of the project when factory-made

units were installed in the laboratory to find any remaining troubles in the production hardware. There were several occasions where it would have been very desirable, for example, to have an event within the ESS halt the microprocessor within the SPUC/DL. We mention in passing that a commercially obtained data communications analyzer proved to be a key instrument in finding certain protocol problems. Ours could record data traffic onto a cartridge tape, stopping after an event of interest; it could also recognize X.25 frame and packet types.

The last few insidious problems in the SPUC/DL turned up in both the hardware and software. The hardware problems were typically sensitivity to "glitches," causing, for example, an erroneous extra DMA cycle. The software problems were usually those of process synchronization and shared resources. The internal architecture of the SPUC/DL, both hardware and software, supports a high degree of concurrency. This increases throughput and minimizes delay, but leads to these internal process synchronization problems. Perhaps more rigorous monitors for shared resources and synchronization semaphores would have paid dividends during the integration phases.

6.2 Particularly successful areas

The problems outlined above led to intense activity as the project neared completion. We were relieved, however, to experience a happy ending; the underlying problems were found and corrected for all observed symptoms.

The climax of our development effort came with the acceptance test for the first 2B ESS—10A RSS field installation. During the 24-hour test, some 60,000 telephone calls were originated on the RSS, processed by the 2B ESS through the SPUC/DL, and properly completed back to the RSS. There were no indications of any problems in the data-link area.

It seems to us that one factor contributing to this success was the attitude that the problems should be *understood* and solved. It might be possible, for example, to stop printing an error message if the message announces some event that seems not to have a harmful effect on the system. We have chosen to explore such events as deeply as resources permit, with gratifying results.

Thus, we feel we have a data-link system essentially free from operational errors. It remains to be seen if our reliability goals will be met.

VII. ACKNOWLEDGMENTS

We acknowledge the considerable contributions of several people to various phases of the project. G. Dobrowski was involved with the

preliminary analysis leading to the architecture of the system. Most of the original SPUC/DL hardware design is by T. Peterson. T. Brinkman assisted with the initial versions of the SPUC/DL microprocessor code. W. McCalla was responsible for most of the original design and strategy decisions in the data-link maintenance area. Much of the data-link maintenance coding was done by R. Atkins, E. Bily, and H. Williams. J. Kent and D. Sikora produced the program for the 2B ESS to diagnose the SPUC/DL. We thank D. K. Ford and S. C. Yuan for their contributions to troubleshooting the hardware, G. Mannon for assistance with modem issues, and D. R. Bierma for continuous attention to testing and administering new issues of both hardware and software.

REFERENCES

1. M. S. Sloman, "X.25 Explained," Computer Commun., *1*, No. 6 (December 1978), pp. 310–27.
2. D. W. Davies, D. L. A. Barber, W. L. Price, and C. M. Solomonides, *Computer Networks and Their Protocols*, New York: John Wiley and Sons, 1979.
3. American Telephone and Telegraph Company, Bell System Technical Reference "Operations Systems Network Protocol Specification: BX.25, Issue 3," Publication 54001, June 1982.
4. Special issue on No. 10A Remote Switching System, B.S.T.J., *61*, No. 4 (April 1982).

AUTHORS

C. E. Ishman, Argonne National Laboratory 1961–1967; Bell Laboratories, 1967—. At Argonne National Laboratory, Mr. Ishman worked on hardware for the zero-gradient synchrotron. Since joining Bell Laboratories he has contributed to many system- and maintenance-software areas on the No. 2 ESS, including call-processing features, both diagnostics and fault-recovery for peripheral units, and conversion of the generic program to the 3ACC processor for the 2B ESS. He is presently a member of the No. 5 ESS System Test Department.

Richard B. Sanderson, B.S. (Electrical Engineering), 1971, Northwestern University; M.S. (Electrical Engineering), 1972, Stanford University; Bell Laboratories, 1972—. Mr. Sanderson has worked on peripheral circuit design for the number 1, 1A, and 2B ESS machines. He is presently Supervisor of the International Circuit Design Group. Member, Tau Beta Pi, Eta Kappa Nu, IEEE.

Louis M. Taff, S.B. (Physics), 1963, Massachusetts Institute of Technology; Ph.D. (Nuclear Physics), 1969, Iowa State University; University of Groningen, Netherlands, 1969–1976; Fermi National Accelerator Laboratory, 1976–1980; Bell Laboratories, 1980—. At the University of Groningen, Mr. Taff engaged in nuclear physics research and became extensively involved with real-time data acquisition systems. At Fermilab, he was a principal in the production of a packaged software system for data acquisition and analysis, now in wide use in high-energy physics. The data-link portion of the 2BE3 generic program was his first project at Bell Laboratories. Mr. Taff is currently working on performance-measurement instrumentation for the number 1A ESS. Member ACM, APS, Sigma Xi.

Donald P. Truax, B.S. (Electrical Engineering), 1966, Michigan State University; M.S.E. (Computer Technology), 1967, University of Michigan; Bell Laboratories, 1966—. Mr. Truax has been involved in the development of fault-recovery, diagnostic, and system integrity programs for No. 1 ESS. Projects have included EADAS and ACD data-link maintenance, and HILO and EPSCS trunk maintenance. Mr. Truax also worked on the PUC and PUC-DL maintenance for No. 1 ESS. He has supervised maintenance projects for both No. 1 ESS and No. 2 ESS. No. 1 ESS projects included PUC-DL maintenance for RSS, ETS, and CCIS and diagnostic improvements for the Remreed networks. Mr. Truax currently supervises the MTSO Maintenance Group for the Advanced Mobile Phone Service project. Member, IEEE, Eta Kappa Nu, Phi Kappa Phi.

Charles T. Tulloss, B.S.E.E., 1974, Tennessee State University; M.S.E.E., 1975, Ohio State University; Bell Laboratories, 1974—. Prior to his work on the 2BE3 data link, Mr. Tulloss has worked in the area of processor-to-processor communications and protocols on the Enhanced Private Switched Communication Service (EPSCS) and the Electronic Tandem Service (ETS) projects. Currently Mr. Tulloss is involved in hardware test and evaluation for No. 5 ESS. Member, Eta Kappa Nu, IEEE Computer Society.

THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 6, July-August 1983
Printed in U.S.A.

Modernization of the Suburban ESS:

Hosting the No. 10A Remote Switching System

By D. W. BROWN,* J. J. DRISCOLL,* F. M. LAX,* M. W. SAAD,* and J. G. WHITEMYER*

(Manuscript received August 31, 1982)

The No. 10A Remote Switching System (RSS) has brought a new dimension to telephone switching in the rural area. The capability to host a 10A RSS was first made available on the metropolitan switches, the No. 1 and No. 1A ESS. The capability has now been extended to the suburban switch, the No. 2B ESS. This article describes the additions and modifications made to the No. 2B ESS to allow it to host the 10A RSS. The discussion also covers the administrative and maintenance features provided and their associated host software implementation.

I. INTRODUCTION

1.1 Definition and basic description

A local Electronic Switching System (ESS) provides the call control for a 10A Remote Switching System (RSS). The 10A RSS acts as a slave executing orders sent to it from the host ESS and reports events, such as line originations, to the host. A major advantage of this type of distributed control is that the complex tasks of call processing can use existing host software and share host equipment and trunking facilities. This sharing of host ESS software makes it possible to easily

* Bell Laboratories.

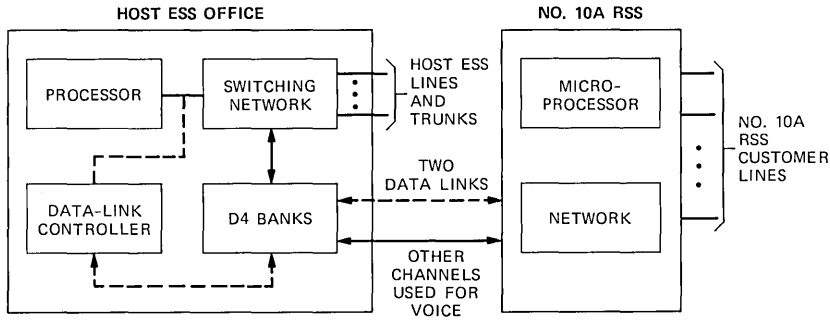


Fig. 1—RSS configuration using digital carrier.

provide RSS lines with the sophisticated features that are offered to host ESS lines.

The major components required for 10A RSS operation include a host ESS, one or more 10A RSS frames, a data-link controller, interconnecting voice channels, and data links. Figure 1 shows this configuration for an application using digital connectivity between the host ESS and a 10A RSS (remote terminal). At this time, the host function has been developed for the Western Electric No. 1 ESS, No. 1A ESS, and No. 2B ESS machines. The data links are used for communication between the host ESS and the 10A RSS and provide the means by which orders from the host are transmitted to the 10A RSS and acknowledgments are returned to the host. Voice channels are used to provide the RSS lines with access to the host network and are selected dynamically. The data link employed for the No. 2B ESS-RSS communication function is a new design utilizing an intelligent Serial Peripheral Unit Controller Data Link (SPUC/DL).* The 10A RSS data-link communication makes use of portions of the X.25 protocol.

The 10A RSS basic frame can serve up to 1024 lines. A companion frame may be added to allow up to 2048 lines to be served by a single RSS entity. The design is such that the basic element of growth can be as small as eight lines. Voice and control communications between the remote and the host can be made over either digital or analog carrier facilities and the range may be as great as 280 miles, depending primarily on the type of transmission facility.

In the event of total carrier system or data-link outage, the 10A

* Acronyms and abbreviations used in this paper are defined at the back of this Journal.

RSS is arranged to automatically transfer to a stand-alone mode of operation, which provides basic telephone service between stations connected to that RSS unit. In the stand-alone mode, special provisions can be made to handle emergency types of traffic such as "911". Details regarding stand-alone operation may be found in an earlier BSTJ article, "Remote Terminal Firmware," by D. A. Anderson et al. in the April 1982 issue on the 10A RSS.

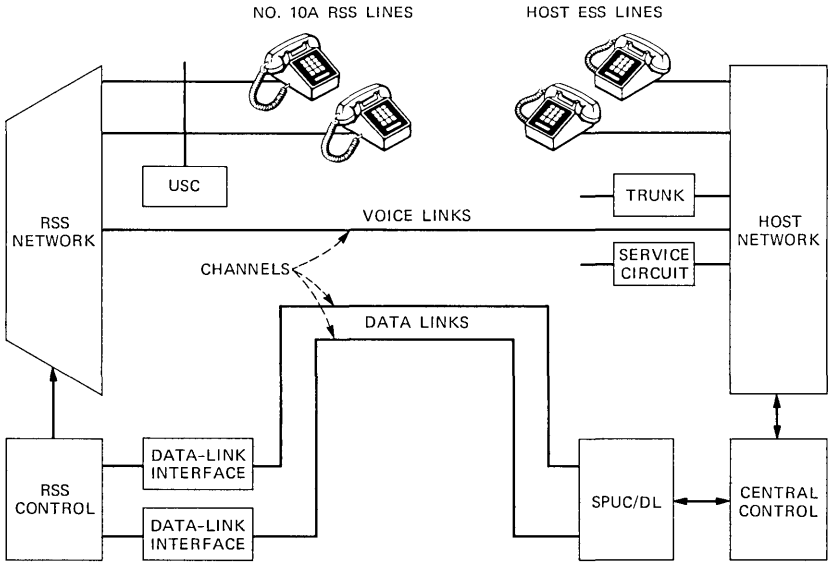
All major units of the 10A RSS are duplicated, and a continuous dialogue is exchanged between the host and remote site concerning the overall health of the system. The basic system philosophy is that the remote unit is a complete slave to the host and merely reports events to the host; the RSS then receives a stream of explicit orders from the host concerning that event. All maintenance procedures must be controlled from the host. Initiation of diagnostics and other functions may be further remoted to a Switching Control Center (SCC).

II. CALL PROCESSING

2.1 Data communications

2.1.1 Hardware overview

Figure 2 shows the 10A RSS—No. 2B ESS host interface and the



ESS – ELECTRONIC SWITCHING SYSTEM
RSS – REMOTE SWITCHING SYSTEM
SPUC/DL – SERIAL PERIPHERAL UNIT CONTROLLER/DATA LINK
USC – UNIVERSAL SERVICE CIRCUITS

Fig. 2—10A RSS—No. 2B ESS host interface.

hardware components involved in the transmission of data between the remote terminal and the host. Communication between the two machines takes place over a pair of low-speed data links that share the same transmission facilities as the voice channels that interconnect the remote terminal with the host. Each data link is placed on a separate transmission facility for reliability. Where carrier facilities are used, the links are assigned to a dedicated voice channel on the carrier system, with each link being assigned to a separate carrier terminal.

Both RSS links are 2400 bps synchronous links. The on-line link is active and carries the entire data traffic between the host and 10A RSS, while the off-line link is maintained in a standby state as a spare. The on-line, off-line status of the links is determined by the host office and is based on error information accumulated by the software responsible for driving the links. At the remote terminal the link is interfaced to the RSS microprocessor through a Data-Link Interface (DLI) circuit. The DLI provides a small amount of data buffering and performs a number of control functions essential to implementing the synchronous link protocol.

At the host, the link interfaces with a functionally similar line interface unit that is part of the SPUC/DL. The function of the SPUC/DL is to provide the control for physically transmitting and receiving data on the links and to provide data buffering for the host office. The data being transmitted and received by the SPUC/DL are buffered on a per-link basis within the terminal. Sufficient data buffering is provided to allow the host to efficiently exchange large blocks of data with the terminal on a schedule that is efficient to the host.

2.1.2 Software overview

The routines that control the data transmission between the remote terminal and host are located in the remote terminal, the SPUC/DL, and the host office. There are two basic functions to be performed: data must be transferred reliably over the link and an interprocess communication system must be provided to allow software processes in the host to communicate with processes in the remote terminal. These two functions are provided by the data-link protocol software and a set of message-routing routines. The protocol provides virtually error-free transmission of data over the link by executing a set of error-detection and error-correction procedures. The message routines allow a process in one machine to direct a message to a process in the other. These two systems are largely independent and bear a hierarchical relationship to one another in the sense that the message routing routines rely on the link protocol routines to accurately transmit data from one end of the link to the other.

2.1.3 Data-link protocol

The protocol routines are executed in the remote terminal and the SPUC/DL. The 10A RSS application uses the link-level portion of the X.25 protocol to control the link. It is a bit-oriented protocol designed for synchronous link operation. To provide for error detection and correction, the data to be transmitted are segmented into numbered blocks termed frames. As frames are transmitted they are sequentially numbered and a cyclic check code is computed over the data in the frame. The frame numbering scheme makes it possible to identify frames for retransmission and to detect missing frames in the received data. As frames are processed at the receiving end of the link, the cyclic check code is recomputed and compared to the one transmitted with the frame. A mismatch indicates that a transmission error has occurred. A positive acknowledgment is returned to the data transmitter for all frames received correctly, and a retransmission request is returned if a frame is received in error.

The protocol software at the SPUC/DL has the additional function of providing link status reports to the host machine. Error conditions such as high transmission error rate, frame acknowledgment timeouts, and loss of data carrier are monitored by the protocol and reported to the host. The transmission error rate is determined from the number of retransmission commands received and sent by the SPUC/DL protocol program. From this data the host data-link state control can take action to remove a link from service if it becomes inoperative or if its throughput is restricted because of excessive data errors.

2.1.4 Message-routing routines

The routines that are responsible for routing data between individual processes in the two machines are executed by the remote terminal and the host processor. These programs assume that data received from the link protocol programs are error free and that any additional error control procedures for detecting transmission errors are unnecessary. These programs are designed to transmit data between buffers associated with client programs in the two machines. A program having data to transmit will load the data into its associated buffer. The buffer will be activated for the message-routing routines and the data will then be transferred to a buffer associated with the destination program in the other machine. The message format is shown in Fig. 3.

2.1.5 Host data transmission

The buffering technique adopted for the transmission of peripheral orders to the remote terminal was designed to be compatible with the structure of the existing host software. The order buffering and mes-

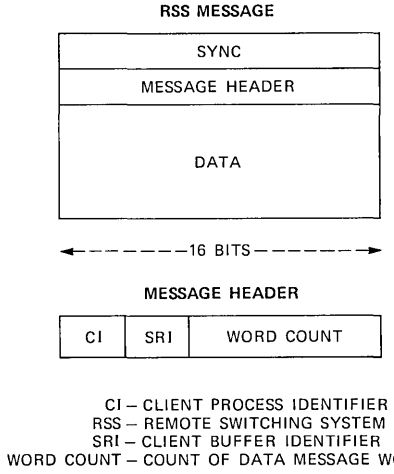


Fig. 3—RSS message structure.

sage transmission must be tailored to the host software structure if the existing call-processing and maintenance routines are to be preserved. A few remarks on the nature of the call-processing programs are necessary to understand the interface requirements.

The host call-processing programs are divided into call segments that process an input from a subscriber or a peripheral circuit to completion in one real-time segment. All the peripheral orders required to process the input are generated by the separate set of input/output (I/O) programs. Peripheral operations in the host or remote terminal take tens or hundreds of milliseconds to execute, which precludes their direct execution within the call segment. During execution, a typical program segment may generate several remote terminal peripheral messages that are destined for different RSSs. In most cases it will also generate a number of orders to be executed in the host periphery in conjunction with the remote terminal actions. The execution routines must be able to coordinate the transmission of the remote terminal orders to the different RSSs. Frequently, it is necessary to execute the remote terminal-host peripheral actions in a predefined sequence where either the host or the remote terminal action must occur first.

The call-processing and maintenance programs are coupled to the I/O programs through a set of I/O buffers that are loaded with the peripheral orders to be executed. All host peripheral orders are buffered in Peripheral Order Buffers (POBs), where they are executed by the POB execution programs designed to handle the timing requirements presented by the host periphery.

A set of Remote Order Buffers (ROBs) in the host machine buffer

the peripheral orders being transmitted to the remote terminals. They are loaded and administered by the call-processing programs in the same manner as the peripheral order buffers. The orders to be sent to the remote terminal are loaded in the ROB via a set of order macros that provide a high-level interface with call processing. When order loading is complete, the ROB is activated and the I/O routines transmit the orders to the remote terminal. An individual ROB may be used to send orders to any RSS. The host can have any number of ROBs pending to send orders to an RSS; however, each remote terminal has a fixed set of eight ROB records that are used to store orders received from a ROB at the host. The ROB records are buffers associated with the peripheral order execution program in the remote terminal that executes the orders transmitted from a ROB.

2.1.6 ROB execution protocol

A rudimentary protocol has been established to coordinate the activities of the call-processing routines at the host with the execution of ROB orders at the remote terminal. Several orders will be grouped together into a single message at the host to be transmitted to the remote terminal. The orders in the message are executed at the remote terminal and upon completion an acknowledgment is returned to the host. The acknowledgment will indicate whether all the orders in the message were successfully executed and must be received by the host before any further actions will be permitted on this call. If the remote terminal encounters a failure in executing an order, the acknowledgment message will specify the failed order to the host and the remote terminal will suspend execution of all remaining orders in the ROB record.

It is essential to receive a positive confirmation on the status of the orders for several reasons. If an order failure occurs, the host fault-recovery routines can be scheduled to clear the call from the system. In addition, the acknowledgment allows the host to correctly sequence any other peripheral actions with the remote terminal orders. When the acknowledgment is received, the host can activate an associated POB or return to a call-processing program to implement the next action on the call. The restriction that additional RSS orders will not be transmitted until the acknowledgment is received also prevents the host from transmitting multiple sets of orders for the same call that would be executed in an arbitrary sequence at the remote terminal.

2.2 RSS network associated data

The processing of RSS calls requires the allocation and management of resources that are physically located at the 10A RSS or shared between the host ESS and the 10A RSS. These resources include the channels that interconnect the host and 10A RSS, receiver off-hook

(ROH) tone circuits located at the 10A RSS, and the 10A RSS network crosspoints. Also, the status of RSS lines is maintained at the host. Several factors were considered in deciding whether to place these functions in the 10A RSS or in the host ESS. These factors are:

1. The effect on service because of the additional time delay if the 10A RSS has to be interrogated to determine line status and to hunt voice channels and network paths.
2. The additional software development required if the host ESS programs have to take a real-time break to interrogate the 10A RSS to obtain a line's status. Host software is structured around data that can be accessed without taking a real-time break.
3. The duplication of development effort that is required to provide the same functions in several host ESS systems.

Factor 3 indicates that the overall development effort would be reduced by placing the line, channel, and 10A RSS network path administration in the 10A RSS. However, the service criteria and the effect on the existing host software were judged to be more important. Therefore, these functions are allocated to the host ESS.

The overall development effort is reduced by making the program and data structure designs independent of the host ESS. Thus, they are highly portable between ESS machines. This section describes the data structures required to administer the RSS resources.

2.2.1 Data structures

Data structures are required in the host ESS memory (call store) to record the status of the RSS facilities and to provide for their administration. To simplify the engineering of the office, these data structures are provided in sizes that correspond to the three basic network sizes of the 10A RSS. Each of these structures is described below.

2.2.1.1 Network block. One network block is provided for every 10A RSS. Its size is fixed regardless of RSS equipage. Among the subblocks contained in the network block are the network map, the channel status blocks, and the remote miscellaneous scan-point status map.

Scan points are provided at the 10A RSS for uses such as alarms, make-busy keys, and stop-hunt keys. The remote unit periodically scans these scan points and, via the data link, reports any changes to the host ESS, which updates the bits in its map to indicate the present state of the scan point.

2.2.1.2 Path memory remote record. A Path Memory Remote (PMR) record is provided for each possible line and channel network appearance. Since the 10A RSS network can be equipped in three different sizes, considerable ESS memory is saved by also providing PMRs in block sizes corresponding to the network size. PMRs contain information about the state of the terminal and a pointer that is used to

point to another memory block (call register or path memory for junctor) involved in the call.

2.2.1.3 Path memory for junctor record. A Path Memory for Junctor (PMJ) record is a block of call store that is associated with a junctor in the 10A RSS network. It is used to store path and terminal information when the junctor is in a network path. A PMJ also contains a state and pointer, which are used to link to another PMJ or to a call register. A PMJ is provided for each equipped junctor. Blocks of PMJs are allocated based on the RSS network size.

2.3 Call-processing strategy

The RSS host call-processing software provides an ESS central office with the capability to supply ESS features to lines served by the remote switching system. Since most of the call-processing functions for RSS lines are performed by the host ESS office, a full family of ESS features can be provided to the remote subscribers. The RSS call-processing software resident in the host ESS provides the means of controlling a remotely located switching system by taking advantage of existing equipment and control capability in the ESS. Firmware in the remote terminal supplements the host call-processing software appropriately. All call-processing control resides in the host ESS and any required actions at the RSS are requested via data-link messages to the RSS. This permits the host to exercise total call control.

2.3.1 Originating call

A line originating in the RSS is first recognized during line scanning performed by the RSS microprocessor. The RSS line-scanning program in the remote terminal recognizes the line off-hook, performs hit timing, and sends an origination request data-link message to the host ESS. If service is allowed, the host marks the RSS line busy and hunts an idle voice channel between the RSS and ESS. It also hunts a path through the RSS network from the originating line to the selected voice channel, and selects a customer digit receiver in the host along with a host network path from the voice channel to the receiver. A Remote Order Buffer (ROB) is then executed to send appropriate data-link messages to the remote terminal to set up the RSS network path and set the line supervision mode to repeat supervision of the originating line over the channel in the dialing (fast repeat) mode. To minimize the dial-tone delay interval, when the above ROB is initiated the host executes orders to its periphery (via a POB mechanism) to set up the host network path between the voice channel and receiver to provide dial tone. That is, the ROB and POB are executed in parallel.

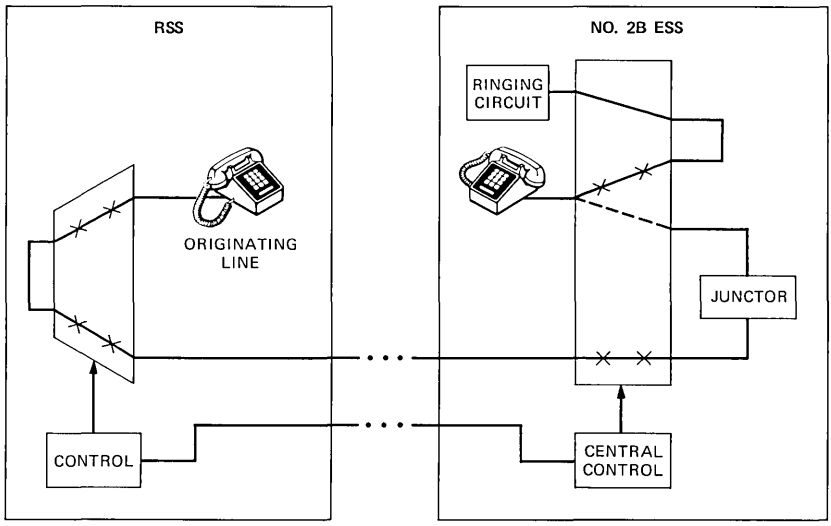
Processing of the call from this time on proceeds basically the same

way as an origination by a host line. Digits are collected and analyzed by the same host software used to process host calls. At the completion of dialing and digit collection, a data-link message is sent from the host to the remote terminal to set the line supervision mode to repeat the supervision of the originating line over the channel in the talking (slow repeat) mode. This mode conserves remote terminal microprocessor real-time capacity.

The RSS originating call, from this point on, is routed and completed normally (excluding terminations to RSS lines) just as non-RSS line origination processing. This originating call configuration is depicted in Fig. 4. Upon answer by the called party or completion of outpulsing, the talking connection is established from the voice channel through the host network. RSS answer timing, billing, traffic, and other administrative functions are all applied and performed by the host just as for non-RSS calls. If the call terminates to an RSS line, special terminating RSS functions are performed, as discussed in the following sections. When either the calling or called parties disconnect, disconnect functions are performed, as discussed in Section 2.3.5.

2.3.2 Terminating call

An RSS terminating call is recognized when the host ESS performs the called number [terminating Directory Number (DN)] translation on digits collected from an originating host line or trunk. RSS lines



ESS – ELECTRONIC SWITCHING SYSTEM
RSS – REMOTE SWITCHING SYSTEM

Fig. 4—RSS originating call.

are distinguished from host lines by special RSS indicators in the terminating line translation output. After the translation is completed, special actions, as with the RSS originating call, are required to set up ringing. The host hunts an idle voice channel to the RSS, hunts a path in the RSS network between the voice channel and the terminating line, and seizes an idle host ringer and audible service circuit with associated host network paths to the voice channel and originating line or trunk, respectively. In addition, a talking path through the ESS network (between the voice channel and the originating line or trunk) is reserved.

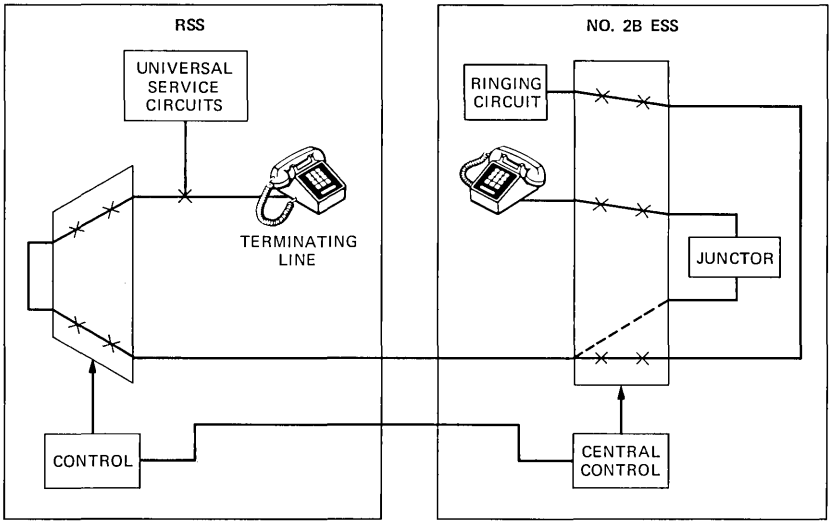
A ROB is activated to send data-link messages to the remote terminal to connect the terminating line to the voice channel and apply ringing to the line. Upon receipt of the data-link orders, the remote terminal selects an idle universal service circuit along with a metallic bus and time slot to provide the type of ringing specified in the data-link message. Supervision of the line is transferred across the voice channel to the host in the fast repeat mode.

Upon successful execution of the ROB data-link orders in the remote terminal, the host executes a POB to set up paths in the host network from the voice channel and the originating line or trunk to its associated service circuit. Power cross and low-line resistance tests are done on the voice channel from the host ringing circuit. The host ringing circuit is then left in a state to monitor ring trip sent by the remote terminal over the voice channel to the host. Actual ringing is applied to the line by the Universal Service Circuit (USC) at the remote terminal; the ESS host ringing circuit does not apply ringing voltage to the voice channel, but is only used to monitor for ring trip. This call configuration, as depicted in Fig. 5, maximizes use of the existing terminating call sequences in the host.

When the called party answers, the remote terminal automatically releases and idles the ringing facilities (USC, metallic access bus, and time slot), relays the ring trip report (off-hook signal) of the line across the voice channel, and sets the supervisory mode to slow repeat.

The host ESS detects answer over the voice channel at the ringing service circuit, tears down the ringing and audible circuit connections in the host, and sets up the talking path that was previously reserved between the voice channel and the originating line or trunk. If the originating line in the RSS terminating call description is actually another voice channel to the same RSS as the terminating line, the call is considered an intra-RSS call and special actions are invoked as described in Section 2.3.4.

Disconnect actions are identical to those for the RSS originating call except for the disconnect timing associated with the terminating versus the originating party.



ESS – ELECTRONIC SWITCHING SYSTEM
RSS – REMOTE SWITCHING SYSTEM

Fig. 5—RSS terminating call.

2.3.3 RSS reverting calls

RSS reverting calls involving a call between two parties on a party line are handled in a manner similar to host reverting calls. However, ringing is provided in a way similar to how it is applied on RSS terminating calls with the exception that two time slots are needed in the RSS remote terminal so that ringing can be applied to both customers. This RSS ringing option, which can be either ac/dc or superimposed, is completely independent of the host ESS office ringing option, or any other RSS served by the same host. The RSS universal service circuit has the capability to provide either ringing option under firmware control.

2.3.4 Intra-RSS call

An intra-RSS call, where both the originating and terminating parties are served by the same RSS, is handled initially as a combination of an RSS originating call and an RSS terminating call. In the No. 1 ESS and No. 1A ESS applications, after answer, the host initially establishes a talking path within its network between the two voice channels. Immediately following the establishment of this talking connection, certain RSS actions are invoked to reswitch-down the call so that the talking connection resides entirely within the RSS network. This releases the ESS network path and voice channels for use on other calls. In the No. 2B ESS application, when answer is received

and both parties are served by the same RSS, the call is reswitched-down immediately without first establishing a talking connection through the host. This sequence will not result in a momentary open interval after the initial talking connection has been established.

The reswitch-down action is initiated when the host hunts an RSS network path within the RSS between the originating and terminating lines. A ROB is activated to send data-link orders to the remote terminal to disconnect both line-to-channel network paths and connect the two lines through the RSS network. The supervisory mode of the RSS lines is set to scan for either a disconnect or switchhook flash, depending on the features associated with each line. Since the intra-RSS connection is entirely within the RSS, a change in supervisory state of the line must be reported over the data link to the host. The sequence of intra-RSS call configurations including reswitch-down is illustrated in Fig. 6.

If a network path in the RSS is not available or if one of the lines is an RSS coin line, the intra-RSS call is not reswitched-down and is connected through the host ESS network. Intra-RSS calls involving coin lines are not reswitched-down in order to utilize host coin-disconnect routines and thus simplify disconnect actions.

The use of various custom calling services or other special services requires a reswitch-up of an intra-RSS call to establish a talking path between the two parties via the host network using two voice channels. This allows existing host software and equipment to be utilized to provide these customer services. The following operations require a reswitch-up operation on an intra-RSS call:

1. A flash by an RSS customer to add on a third party
2. A terminating call to one of the two parties of an intra-RSS call that has the call waiting/terminating feature
3. A busy verification test by an operator of one of the two parties of an intra-RSS call.

When the host determines that a reswitch-up function must be performed, for any of the reasons given above, the host seizes two idle voice channels to the RSS and hunts a path between them in the host network; the host also hunts a path between the two voice channels and both lines in the remote terminal. A POB is executed in the host to set up a talking connection between the two voice channels followed by a ROB to send data-link messages to the remote terminal to disconnect the intra-RSS talking connection, connect each line to a voice channel, and set the supervision state of each line to the talking mode (repeat supervision of the line over its respective voice channel). Once the intra-RSS call is reswitched-up to a talking connection via the host ESS network, the original service requested can be provided just as it would to a normal line-to-line connection of two host lines.

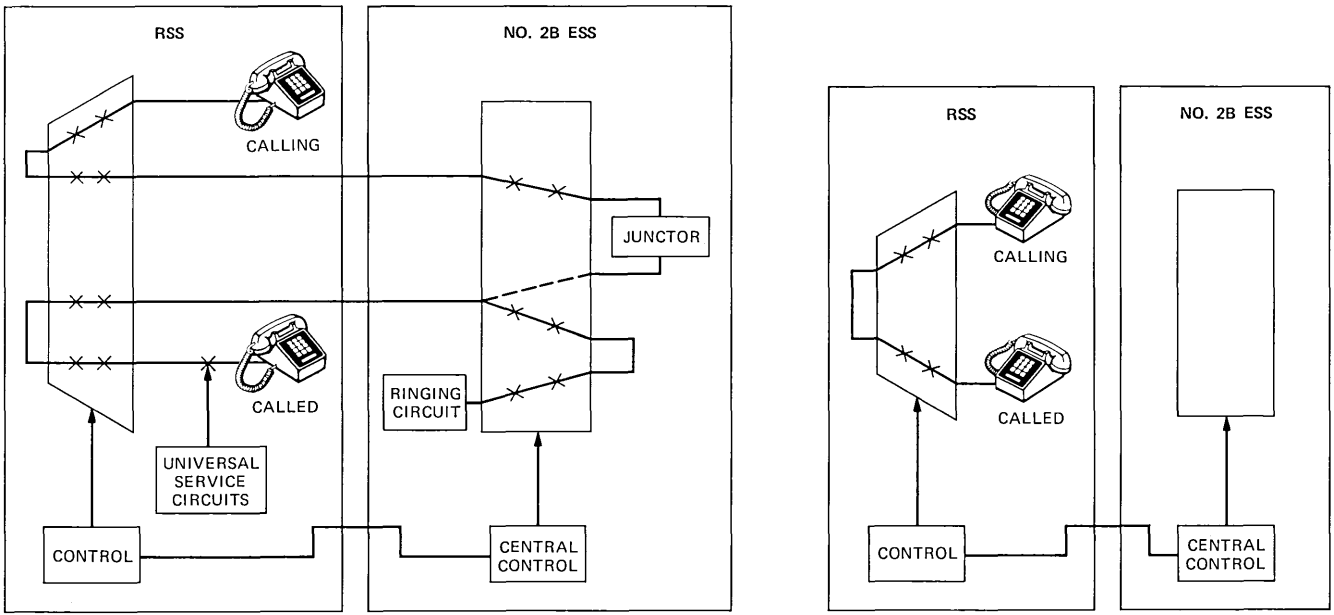


Fig. 6—Intra-RSS connection sequence. (a) Ringing connection. (b) Talking connection.

2.3.5 Disconnect functions

Disconnect actions for RSS calls are a function of the particular call configuration involved, i.e., intra-RSS calls or RSS calls through the host network. For call configurations involving both RSS and ESS paths, the ESS disconnect programs control the call-disconnect actions. The same disconnect sequence that is performed on ESS host lines is used on RSS channels that terminate on the host line network. This disconnect control strategy provides the ability to centrally recognize an RSS channel during normal host-disconnect processing. This recognition occurs when ESS programs perform a restore-verify action on the RSS channel. At this point in the host-disconnect processing, unique host RSS disconnect modules are invoked to disconnect the network path of the RSSs. The normal host-disconnect program and RSS-disconnect module then autonomously complete their respective disconnect actions. The only common resource between the two control programs is the RSS channel. The ESS host-disconnect program administers the host end of the channel on its network, and the RSS host-disconnect program administers the RSS end of the channel on its network.

In the case of an intra-RSS call, where the call configuration involves a connection totally within the RSS network and no channels or ESS network paths are involved, the RSS lines are supervised in the 10A RSS. Hits, flashes, and on-hooks are detected by the RSS; flashes and on-hooks are reported to the ESS host via data-link message. These messages are routed to unique RSS disconnect control modules in the host for proper processing. In the case of an on-hook, these programs perform the proper disconnect timing and then execute ROBs to disconnect the intra-RSS network paths and idle the lines involved. The intra-RSS call is reswitched-up on receipt of a flash message (as discussed in Section 2.3.4).

2.4 Database integrity

In an electronic switching system, the status of resources and telephone calls is recorded in temporary memory. This data can become mutilated because of program bugs, hardware errors, or program design errors, resulting in the loss of resources or system degradation. The problem is further complicated by RSS because parts of the new data structures in the host ESS are duplicated in the remote terminal. The new structures are:

1. PMRs for lines
2. PMRs for channels
3. Network map
4. Remote order buffers (ROBs)
5. Remote miscellaneous scan point map.

The actual data stored in the two copies of these structures are not identical in all cases since the host and remote terminal do not perform identical functions. For example, the state stored in a line PMR at the remote terminal represents the supervisory state of the line (origination, repeat supervision onto a channel, high and wet, etc.), whereas the state in the host PMR represents both the status of the line (idle, busy, maintenance) and the type of path memory configuration, as discussed previously. However, there is a mapping between the two sets of states in that the host-line state implies the possible supervisory states of the line. Deviations from this mapping should only be due to the time lag of the data link.

To ensure the integrity of the data structures, the first step is to prevent as many errors as possible. RSS software applies many of the techniques that have been successfully used in other ESS projects to avoid potential causes of errors. Some of these are good documentation, standardized program interfaces, structured design, structured programming, a high-level programmer's language, and a standardized data definition language. In addition, access to the new data structures introduced into the host is limited to the administrative programs that have the responsibility for that particular database.

The second step is to make the programs as error tolerant as possible since data errors will still occur. The main technique for this is defensive programming. The degree to which a data error is propagated through the system depends upon how the programs use the data. In order to have a minimal effect upon the system, programs should account for bad data. Some specific types of defensive coding techniques are:

1. Range checks on data to prevent overindexing tables
2. Accounting for all possible subroutine return code values
3. Use of symbolic definitions for data values
4. Accounting for all possible program inputs
5. Invalid data value checks.

Despite the preventive and defensive techniques that are employed, errors can still occur in the data. Programs are required to detect these errors and restore the facilities to the proper condition to avoid system degradation. These audit programs are responsible for detecting and correcting data errors and the initialization programs are responsible for restoring system facilities when the degradation is severe enough to cause major system degradation.

2.4.1 Audit programs

The integrity of the data structures is checked and corrected by a set of audit programs. Each audit program is individually tailored to a specific data structure or group of data structures and determines if

the data items follow certain established rules. If the checks fail, the audit programs make appropriate corrections to all resources (both software and hardware) associated with the particular error and print error messages on the teletypewriter. The audit programs also aid in the initialization of the data structures.

The audit philosophy adopted for RSS is that each entity will maintain the integrity of its databases independently. If the host finds a discrepancy in its database, it corrects the problem by resetting the state of all host resources involved and instructs the remote terminal to put its facilities into a known (usually idle) state. If the remote terminal finds a discrepancy in its database, it sends a message to the host and the host audit programs initiate the actions given above.

Thus, there are three classes of audits associated with the RSS system:

1. Host audits that maintain the internal integrity of the data structures in the host
2. Remote terminal audits that maintain the internal integrity of the data structures in the remote terminal
3. Audits that guarantee that the host and remote terminal data structures are consistent.

Audit classes 1 and 3 are discussed below.

2.4.1.1 Host audits. Existing host audits are extended to include the new data structures and new data values introduced with RSS. The main audit modifications are for the RSS path memory, the RSS network map, channels, and ROBs.

The RSS path memory and network map are audited by making the following checks:

1. Point-to-point-back checks are performed between PMRs and PMJs.
2. Point-to-point-back from a PMR or PMJ to a call register are performed if linkage to a call register is indicated.
3. The junctor busy-idle bit in the network map is checked to ensure that it is idle if and only if the PMJ is idle.
4. Each of the other network map bits that is marked busy is checked to guarantee that it is in a valid path.

ROBs are audited by periodically rebuilding the idle link list and by timing ROBs associated with transient call records. If a ROB remains busy for an extensive length of time, the ROB and any associated call register are idled. All paths and circuits are also idled.

The corrective action taken by the host audits is to idle facilities (hardware and software) in the host and to send orders to the remote terminal to cause the facilities at that end to be idled.

2.4.1.2 Synchronization between host and remote terminal. The problem of maintaining the data structures in the host and remote terminal in

synchronization is greatly simplified by taking advantage of the normal system operation. The remote terminal updates its data in response to orders from the host. Bits in the remote copy of the network map are marked busy or idle in response to orders to set up or tear down network paths. Thus, no network map audit is required between the host and remote terminal since the host does the hunting and idling of paths and the remote copy will tend towards the proper state even if it does temporarily get out of step.

Similarly, ROBs are controlled from the host end and normal operation will result in the remote copy being brought back into step with the host.

The remote terminal audits check that all equipped lines have supervision turned on. Any equipped lines that are unsupervised are reported to the host via a data-link message. If the host audits determine that the line state really calls for supervision to be on, the line and any associated resources are idled.

Periodically, the host sends a copy of its version of the remote scan point map to the remote terminal, which overwrites its map with the host's data. If the hardware state of the scan point differs from the map, the normal scan program will detect this as a change and report it to the host, resulting in both copies being brought back into step.

2.4.2 Remote terminal initialization

System initialization programs are responsible for correcting errors that prevent the system programs from cycling correctly. The initialization programs are usually executed as a consequence of errors being detected by the processor check circuits that monitor the sanity of the system operation. In the remote terminal, the primary checks for monitoring proper program operation are the system sanity timer, which monitors the main program cycle time; the write protect circuitry, which prevents illegal writes into program store; and certain peripheral error checks, which detect attempts to access unequipped areas of the periphery. If any of these errors are detected, it is indicative of an error in the system database. The method of recovery is to initialize a segment of the data and then return to the normal program cycle.

Since the initialization process inherently destroys a portion of the call-processing data, a corresponding set of calls will be lost, and it becomes a requirement for the initialization program to release the peripheral circuits associated with these calls. Any network links, channels, or service circuits employed on these calls must be idled by the initialization program before a return to normal system operation is begun. This is accomplished by releasing all the circuits that are marked idle in the initialized database.

Whenever an error is detected by a fault-detection circuit, a processor interrupt is generated that executes the fault-recovery programs. Various fault-recovery actions are taken, depending on the type of errors detected and their frequency.

The amount of data that is initialized on the first error is small. As successive errors are detected, the severity of the initialization is increased with the consequent loss of progressively greater numbers of calls. The ultimate action is to initialize the entire database and restore the system to an idle state. The goal of handling the fault recovery in stages, with increasingly more severe initializations, is to restore the system to a working mode with the loss of a minimal number of calls.

At either the remote terminal or the host, there are three fundamental levels of initialization: a minimal clear, a transient clear, and a stable clear. A minimal clear involves the initialization of the variable data associated with the active processes in the system and has the potential of disrupting, at most, several calls. A transient clear will initialize all the data in the system that is related to any call in progress. All calls in the process of being established or disconnected will be lost; however, calls in a stable talking state will be preserved. The final state of initialization is a stable clear where the entire database is reinitialized and all calls are lost.

For the RSS system, the initialization process is somewhat more involved than normal because the host and remote terminal databases are interrelated and an initialization level (phase) in one machine affects the database of the other system. Although the host machine is responsible for the control of the remote terminal on a call-related basis, the operation of the processors in the two machines is fairly autonomous with respect to their instantaneous activities. This is the situation for fault detection where the two systems are entirely independent. Each machine is responsible for initializing and carrying out its own fault-recovery actions and the level of the accompanying initialization will be solely determined by the conditions within the machine that detected the error. In effect, either machine is able to initiate any level of initialization on its own database independently of the other. However, as part of the initialization procedure, the hardware and software within the two machines must be synchronized so that the databases reflect a consistent set of calls. The calls that were destroyed in one machine must be reported to the other so they can be cleared from that system also. For example, a host-transient clear will destroy a number of calls that involve remote terminal lines. These calls must be cleared in the remote terminal so that periphery and call records are in agreement with those in the host.

The exact procedure for synchronizing the two machines will depend

on which system has initiated the phase. Since the host is in charge of call control, its call records are regarded as the master copy and the remote terminal state is brought into agreement with its set of records. The host is therefore in charge of synchronizing the two machines.

Whenever a phase occurs in the host, the synchronization procedure is straightforward. The host will initialize its database and periphery and will then send initialization orders to the remote terminal to bring it into agreement with its updated records. When the remote terminal undergoes an initialization, it reports the level of the initialization to the host. In some instances, on a stable clear, for example, this is sufficient information to allow the host to initialize its database. In the case of a transient clear, it is also necessary to transmit a map of the lines that are in a transient state in the remote terminal. From the data specifying the initialization level and the map of transient lines, the host is able to update its call records and periphery. Once this is accomplished, it will conduct an initialization of the remote terminal in the same manner as for a host-initiated phase.

During a high-level initialization, interactions between the host and the remote terminal must be modified. For example, during the resynchronization of the host and remote path memory transient databases, it is necessary to prevent these databases from being accessed by call-processing routines. To implement this capability, the RSS is modeled as a finite-state machine with ten possible states. The state of each remote terminal together with a list of activities allowed in that state is maintained by the host in a database. Software, which functions differently based on the state of the remote terminal, checks this database to determine what action is appropriate in the given RSS state.

III. MAINTENANCE ENHANCEMENTS

Extension of telephone communication service through the 10A RSS brings with it an associated extension of maintenance capabilities. An obvious maintenance need relates to the SPUC/DL and data-channel facility between the No. 2B ESS and No. 10A RSS processors. Integrity and maintenance issues for this subsystem are dealt with in the companion article of this series, "Adding Data Links to an Existing ESS," by C. E. Ishman et. al.

Another area requiring maintenance enhancement is related to the addition of voice channels to the network architecture of the combined systems. Unlike other "links" of the No. 2B ESS network, the voice channels are nonmetallic, largely external to the central office environment, and are maintained by a separate craft force. These factors require the design to deal with issues of availability, transmission and

noise performance, and trouble sectionalization. The channel maintenance software package, developed to address these issues, is described in the following sections.

Other maintenance additions, for telephone customer loop testing, are necessitated by the remoteness of the RSS and the nonmetallic network upon which these loops terminate. Testing capabilities, similar to those available for host-terminated lines, must be provided for RSS customer loop trouble detection and resolution. Such capabilities are described in the following sections on line maintenance. Finally, as a means of achieving test hardware economies in the RSS, the host diagnostic capabilities were expanded to test Dual-Tone Multifrequency (DTMF) receivers at the RSS as well.

3.1 Channel maintenance

Basic needs for the effective maintenance of RSS voice channels may be divided as follows:

1. The system must provide routine, automatically initiated tests that periodically diagnose individual channels to detect performance-affecting faults—particularly faults characterized by progressive and marginal degradation.
2. The above set of tests should also be initiated by call-processing programs in instances where in-process integrity checks indicate errors that could be caused by faulty channels.
3. The resolution of some hardware faults will require manual intervention. This is accomplished by: (a) allowing the above diagnostic tests to be executed upon demand, and (b) providing dc and ac test access to each channel for voltmeter-type testing. This latter facility has been provided through an enhancement of the existing central office Trunk Test Panel (TTP).

3.1.1 Channel diagnostics

The channel diagnostics, which play a major role in fulfilling all three needs, have been designed with a "start-small" philosophy; the sequence of tests is selected so as to confirm basic functions first, and then to build upon this knowledge in carrying out subsequent evaluation. The following sequence of tests is terminated when a failure is detected at any step:

1. Power cross test
2. False cross or ground test
3. Supervision test
4. Restore-verify test
5. Low line resistance test
6. Dial pulse test

7. ac far-to-near test
8. ac near-to-far test.

The power cross test, shown schematically in Fig. 7a, uses the same detecting circuit as do customer lines and is designed to detect crosses to central office battery or commercial power in the cabling between the No. 2B ESS switching network and the carrier terminal. By performing this test first, potential damage to other test circuits is avoided. The False Cross or Ground (FCG) test cannot use the existing test method since it is overly sensitive to the capacitative load seen at the input of the carrier's channel unit. Instead, the existing continuity and polarity test circuit is connected, as shown in Fig. 7b, to detect the existence of conductor-to-conductor and conductor-to-ground shorts (up to approximately 2000 ohms) in the same cabling.

The supervision test verifies the ability of the channel to pass on-hook and off-hook signals from the RSS remote terminal to the No. 2B ESS host. With the channel idle, on-hook supervision is first confirmed by applying loop battery toward the channel with the continuity test circuit (as diagrammed in Fig. 7c). After sending an off-hook order to the RSS (via the data channel), the central office current detector is scanned for the expected off-hook.

The restore-verify test, in a manner similar to that used with customer lines, verifies that the supervisory attending element can be reconnected to the channel and can detect the off-hook signal associated with a channel "origination." This is accomplished, as shown in Fig. 7d, by causing the RSS to transmit an off-hook signal toward the host when the host has its line attending element connected to the channel.

The low line resistance test is still another carry-over from host-line testing. Here, the objective is to detect the existence of high-resistance (up to approximately 15,000 ohms) crosses from conductor to conductor and from conductor to ground. While the intent of making such a test on a customer line is to prevent false ring tripping, the benefit to channel testing is to better characterize the nature of a fault. This test connection is shown in Fig. 7e.

Since originating RSS customers make use of central office digit receivers, the voice channels must be capable of conveying dialed digits, which are transmitted on the channel as a series of supervisory transitions. The dial-pulse test performs this function by: (1) connecting the central office end of the channel to a digit receiver, and (2) requesting the RSS to transmit a digit "0" (ten dial pulses). The test passes if the digit receiver shows that ten dial pulses have been received. The associated connections are shown in Fig. 7f.

The final two tests on the channel form a gross verification of its two-way transmission capability. The ac far-to-near test, shown in Fig. 7g, connects a source of milliwatt level 1-kHz tone at the remote

end and a tone detector at the central office end. The complementary near-to-far test, shown in Fig. 7h, verifies transmission in the opposite direction and employs a similar technique. Since the tone detectors used in these tests are sensitive down to −30 dbm, the diagnostic results cannot confirm adherence to rigid (±1 dB) limits. The Remote Office Test Line (ROTL) feature has therefore been enhanced to perform accurate transmission measurements on voice channels as well as on trunks. This capability is discussed in the following section.

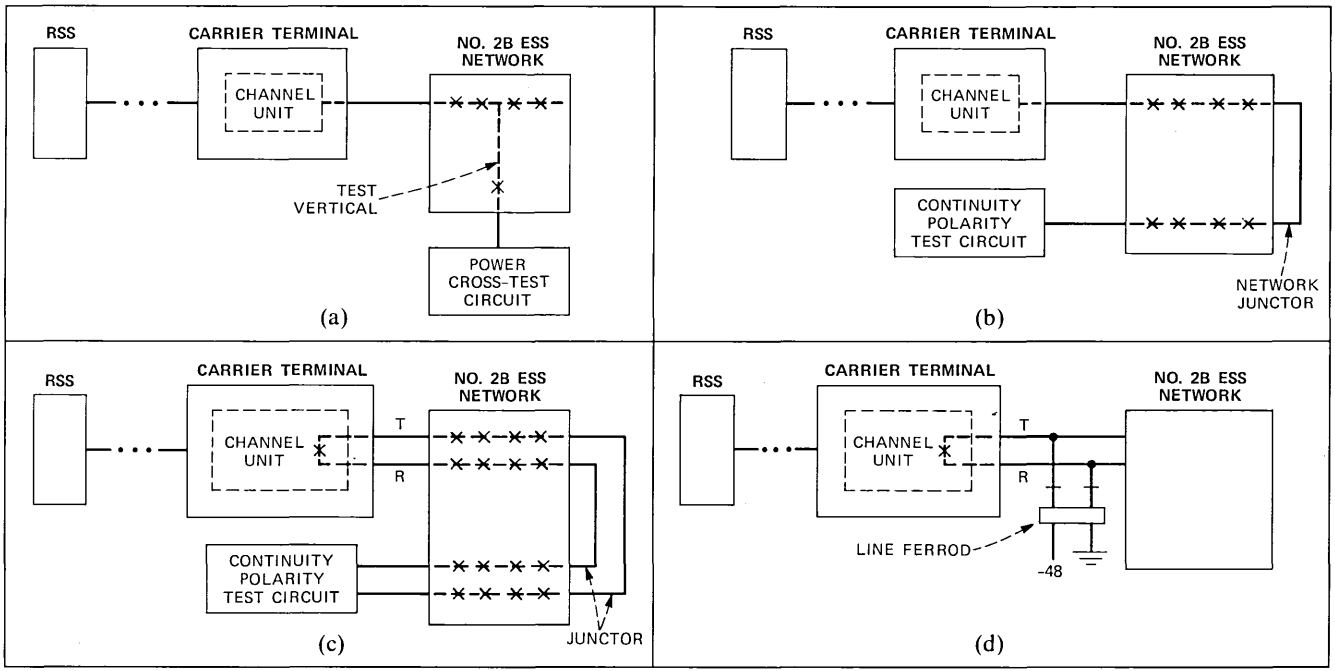
3.1.2 Automatic channel transmission testing

The ROTL feature, first implemented in the No. 2-EF-1 generic, was designed to allow automatic transmission testing of trunks by the Centralized Automatic Reporting on Trunks (CAROT) processor. Since the RSS channels have a functional role similar to that of local interoffice trunks, enhancement of the ROTL feature was selected as an efficient method for channel transmission testing. The resulting design, as implemented in the 2BE3 generic, proceeds along the following typical sequence in the testing of a channel:

1. The CAROT center connects to the ROTL access (incoming) port via the standard Direct Distance Dialing (DDD) network.
2. Priming data specifying the test parameters is sent from the CAROT, through the ROTL access port, to a Multifrequency (MF) receiver. This digit receiver is accessed via a normal central office connection from the ROTL test port, as shown in Fig. 8.
3. Upon receipt of the channel identity, the central office releases connections to the MF receiver, establishes a connection between the ROTL test port and the selected channel, and requests the RSS to connect a transponder to the remote end of the selected channel.

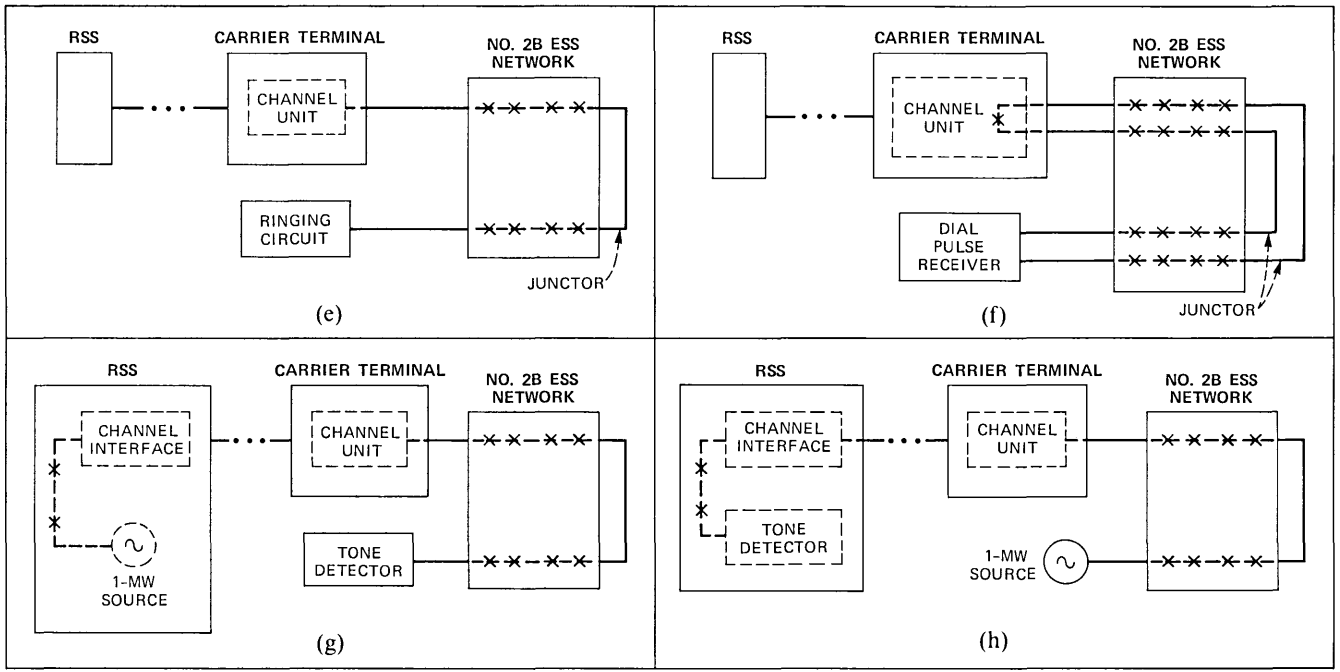
During this sequence of events, considerable data "handshaking" takes place between the CAROT and the No. 2B ESS—much of this communication in the form of bursts of "test progress tone" sent by the ROTL circuitry to the CAROT. The end result is a composite connection, as shown in Fig. 9, which permits the RSS transmission measuring transponder (termed a miniresponder) to interact with the CAROT in performing transmission measurements in either direction on the channel. The CAROT may then restore or remove channels from service based on the test results.

Since the CAROT-ROTL system can make loss and noise measurements accurate to 0.1 dB, this design provides a low-cost means of maintaining voice channel transmission characteristics within acceptable limits. Also, because the RSS miniresponder can be accessed in a similar fashion using a portable ROTL System Test Set, the same high-resolution measurements are available to the maintenance craft on a demand basis.



ESS – ELECTRONIC SWITCHING SYSTEM
RSS – REMOTE SWITCHING SYSTEM

Fig. 7—Test configurations of channel diagnostic: (a) Power cross test. (b) False cross/ground test. (c) Supervision test. (d) Restore-verify test.



ESS – ELECTRONIC SWITCHING SYSTEM
RSS – REMOTE SWITCHING SYSTEM

Fig. 7 (Continued)—Test configurations of channel diagnostic: (e) Low-line resistance test. (f) Dial pulse test. (g) AC far-to-near test. (h) AC near-to-far test.

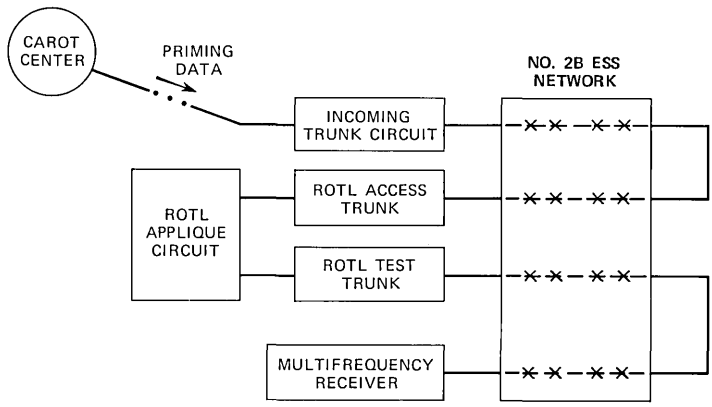


Fig. 8—Initial CAROT-ROTL connection.

3.1.3 In-process error detection

While routine testing is an essential ingredient in the channel maintenance package, means must also be provided to deal with channel-related failures encountered during the processing of a telephone call. Such a facility will properly dispose of channels with transient faults (which are not detected during periodic exercise) and channels with newly occurring faults.

Once a channel is implicated by a failure in call processing, automatic routines dispose of the channel as follows:

1. A diagnostic test is run on the channel and, if the test fails, the channel is removed from service (subject to previously specified numerical limits).
2. If the diagnostic passes or cannot be run because of resource blockage, a report of the incident is made to error analysis programs running at the RSS.
3. The error analysis routines, which accept failure input from the remote terminal as well as the host, evaluate the failure history of each channel. This evaluation is carried out using two algorithms: a peer group comparison and a "quick check."
4. If the peer group analysis indicates that a given channel's error rate is significantly higher than that of its peers, that channel will be removed from service (subject to the specified maintenance limits).
5. If the error analysis indicates a "quick check" failure (three successive channel errors occurring on the same channel), channel diagnostics are run on the suspected channel. Depending on the

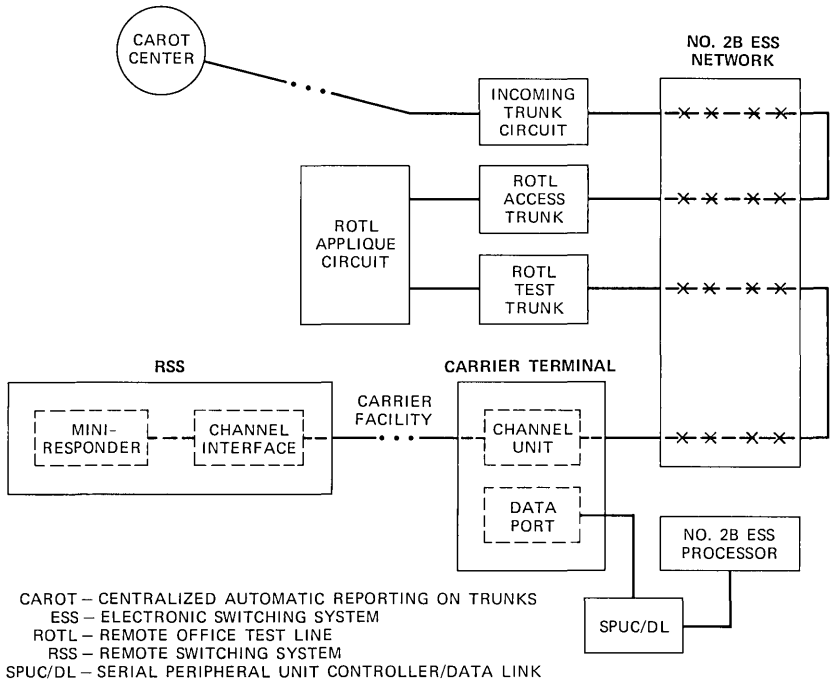


Fig. 9—Channel testing with CAROT and ROTL.

success or failure of the diagnostic exercise, the action taken is the same as that listed in (1) and (2) above.

3.1.4 Manually controlled channel testing

As previously mentioned, the ROTL capability for channel (a) testing may be used automatically (via CAROT) or manually. In addition, to provide direct metallic access to the channel and control of its associated circuit states, the central office trunk test panel programs have been modified to provide test functions similar to those available for trunks.

A typical sequence, which permits a loop-around connection of two channels, is itemized below. Such a connection is used in two-way transmission loss measurements on channels.

1. Using the panel-mounted telephone set at the trunk test panel, digits are dialed that direct a specified channel (a) to be connected to a source of milliwatt tone (i.e., a 102-type test line) at the RSS. This connection, shown in Fig. 10, permits the TTP transmission measuring set to determine the loss of the channel in the far-to-near direction.

While this channel is accessed, existing keys at the TTP may be used to change the circuit state of the channel at the RSS. (These

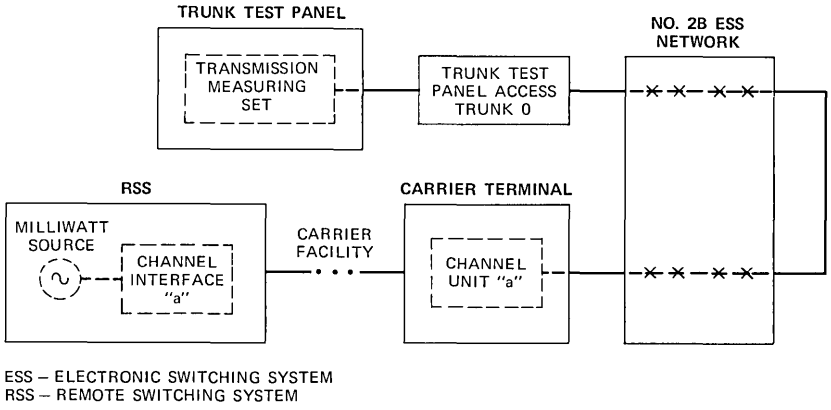


Fig. 10—Measurement of channel far-to-near loss.

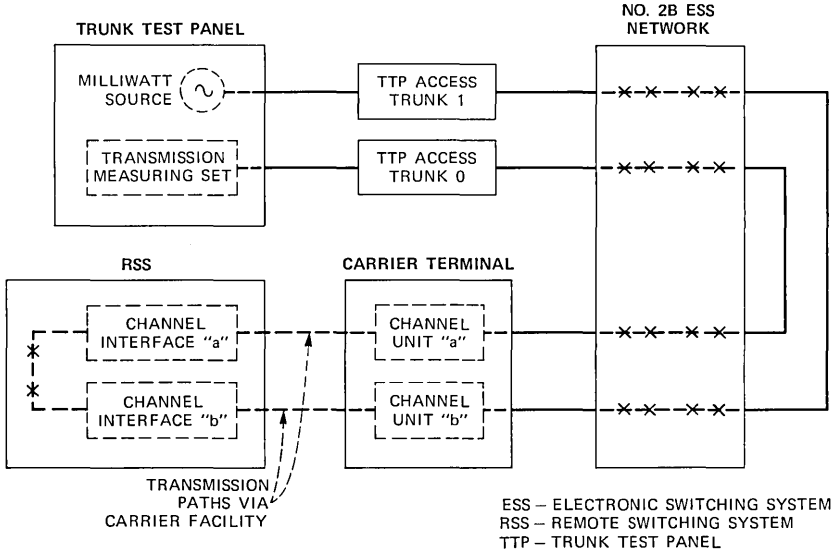


Fig. 11—Measurement of channel near-to-far loss.

states correspond to supervisory conditions, loss pads in and out of circuit, and the connecting of a balanced terminating network.)

2. Dialing a different test code and channel identity will cause a second channel (b) to be connected between another TTP access trunk and a loop-around connection at the RSS. This connection, shown in Fig. 11, has produced a configuration in which both ends of a two-channel transmission path are connected to the TTP. Now with a source of 0 dbM connected to the second channel and a transmission-

measuring set connected to the first, the near-to-far loss on the second channel may be determined.

3. When any channel is released from the TTP, several options are possible, depending upon the channel's previous status and the method of TTP disconnect. For example, if the channel had initially been out of service, TTP release will initiate a diagnostic test of the channel, restoring it to service if the diagnostic passes. If an initially idle (in-service) channel is released with the "make busy" key operated, the channel is unconditionally removed from service.

3.2 Line maintenance

Like channel maintenance, line maintenance may be subdivided into capabilities that: (1) by periodic, automatically initiated testing, attempt to detect progressive hardware degradation before service is affected; and (2) by allowing demand-type human-controlled testing, enable the maintenance craft to resolve the nature and location of faults. Automatic Line Insulation Testing (ALIT) occupies the first of these categories and, like its counterpart for host-terminated lines, detects high-resistance crosses and grounds on metallic pairs between the RSS and the customer premises. The second category of features includes an interface to the Local Test Desk (located in a centralized repair service bureau) and the station ringer test, which verifies the correct operation of the subscriber's station set.

3.2.1 Line insulation testing

If it were not for the nonmetallic network in the RSS remote terminal, line insulation testing could be performed by the host No. 2B ESS. Instead, the remote terminal is equipped with its own insulation testing circuit and the means to make a metallic connection between this circuit and all customer line terminations.

As shown in Fig. 12, the RSS ALIT circuit accesses the customer line via the Metallic Access Bus (MAB), the Line Test Access Bus (LTAB), and the Universal Service Circuit (USC), the latter in the "bypass" state. RSS firmware has exclusive control of this connection and testing sequence once a line and test parameters are specified via data order. That is, after the RSS receives a line test request from the host, the remote terminal selects an idle USC, connects the MAB, performs the test, disconnects from the line, and returns the test results in an acknowledgment message over the data channel. The test parameters received in the requesting message are similar to those specified for host line testing and permit testing sensitivities ranging from 80 kΩ to 5 MΩ. In addition to the line identity, the requesting order also includes a specification of the test mode. The mode, an

operating company and craft option, selects the extent of the testing performed on an individual line. The choices are:

1. A foreign EMF (electromotive force) test, which looks for false power crosses to either conductor,
2. The TRG (tip-ring to ground) test, which measures resistance between the conductors (tied together) and ground,
3. Leakage test, which measures resistance between the conductors,
4. A "general" test, consisting of a sequential application of all the preceding tests.

RSS line insulation testing is included in the automatically initiated sequence and follows, RSS by RSS, the insulation testing of host lines. While the testing of RSS lines could have been done concurrently with the testing of host lines (since separate and dedicated hardware is involved), such a design would have increased the complexity of teletypewriter output messages, which identify line failures. By completing the testing of all lines in one entity (i.e., the host, or one of the RSSs) before proceeding to test lines in the next, a single "header" message serves to identify the entity for all line failures that follow. This has the added benefit of grouping those reports corresponding to their geographic locations.

3.2.2 Local test desk

Demand-type line testing of host and RSS lines may be performed from a test cabinet, located within the No. 2B ESS central office, and from a Local Test Desk (LTD) at some remote location. To accommodate the variation of metallic and carrier facilities between the LTD and the host and between the host and an RSS, a number of options have been provided. The simplest case, from a design point of view, exists where there is a low resistance (less than 2600 ohms) metallic path from the LTD to the RSS customer's telephone. As shown in Fig. 13, the connection from the LTD to the host uses the standard No. 2B ESS—LTD incoming trunk circuit and a dedicated metallic facility between the RSS and the host. This metallic connection is completed in the RSS by connection through the Metallic Line Access Port (MLAP), the USC, and the MAB. LTD testing, which consists of various voltage and resistance measurements, may then proceed over the established connection.

Since such low-resistance facilities are expensive and rarely found in the rural and suburban environment, an alternative implementation is provided. This arrangement takes advantage of the Remote Testing System (RTS), which was previously provided to work over nonmetallic facilities between the LTD and the central office. The resulting design employs a multifrequency telemetry Remote Line Test (RLT) unit at the RSS. As shown in Fig. 14, the microprocessor-controlled

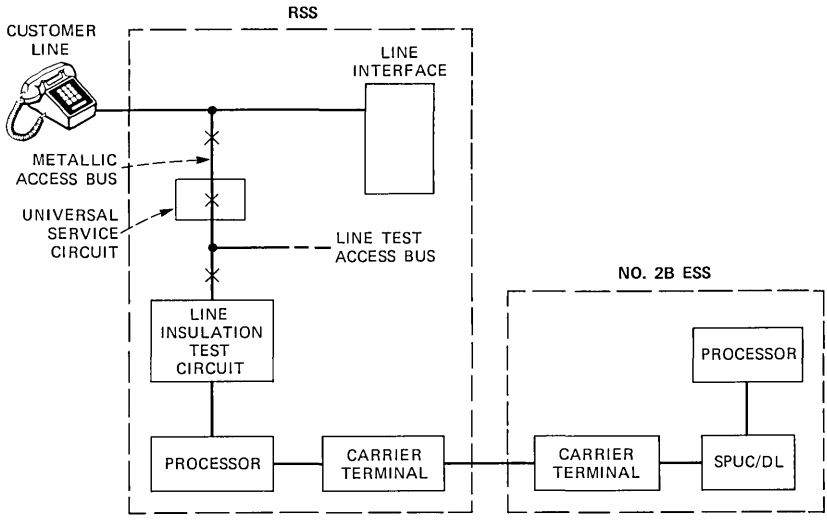


Fig. 12—RSS line insulation testing.

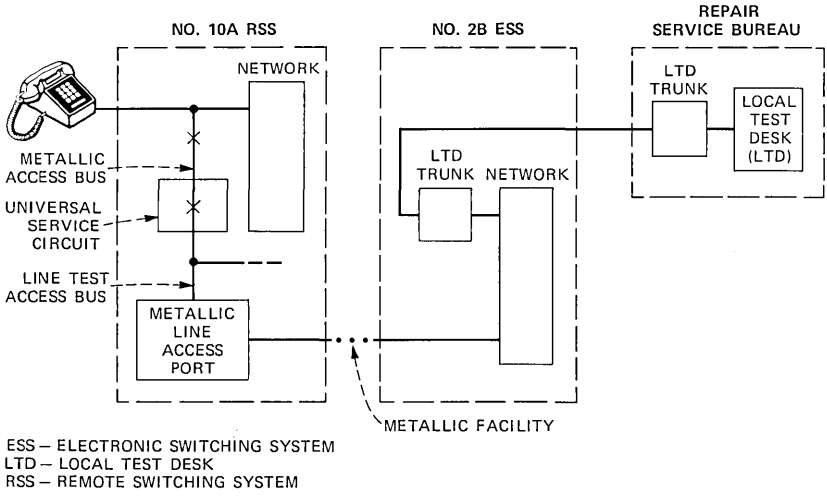


Fig. 13—Metallic option for the local test desk.

RLT has metallic access to the customer line via the MLAP, USC, and MAB. The connection of the LTD and RLT provides for test requests being sent to the RLT and for test results being returned to the LTD. Supervisory signals from the LTD (e.g., disconnect) are

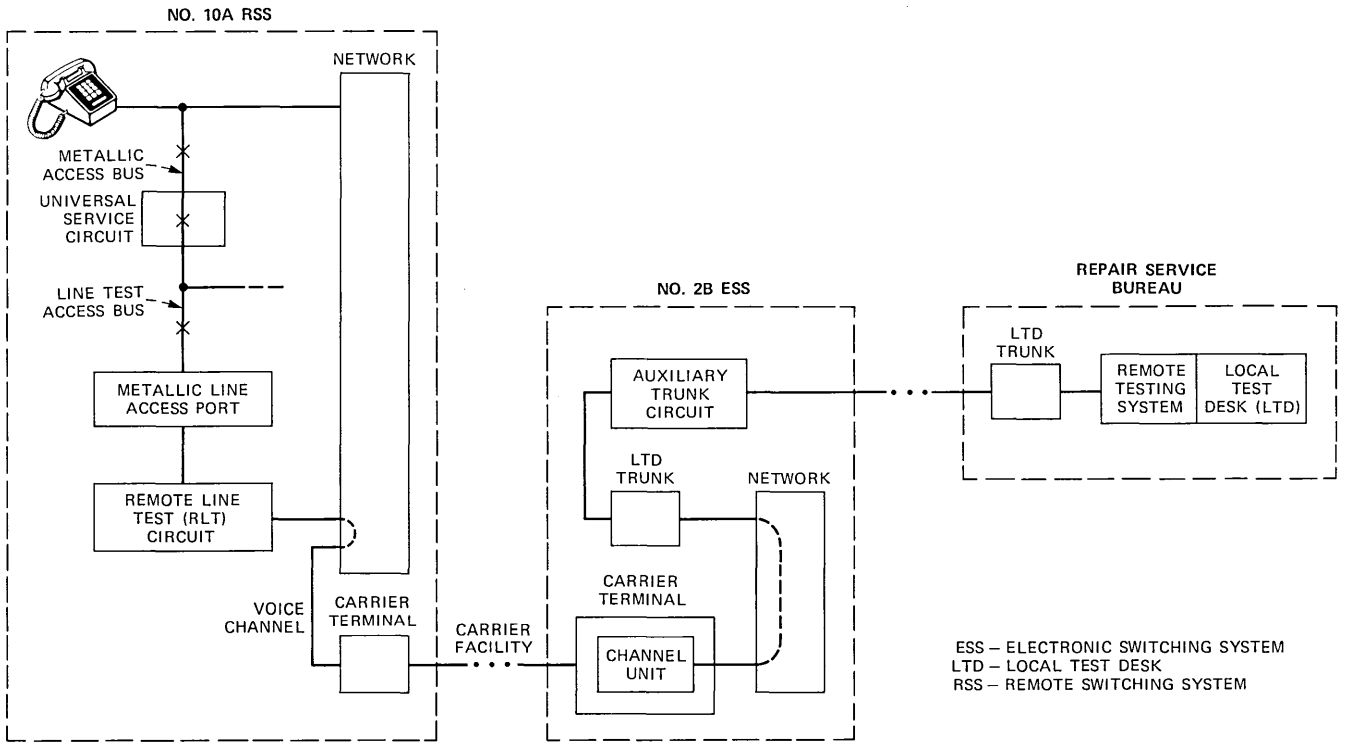


Fig. 14—RLT option for the local test desk.

intercepted by the auxiliary trunk circuit in the No. 2B ESS, since the host is responsible for maintaining and tearing down the entire connection. The RLT and the auxiliary trunk circuit at the host have therefore functionally replaced the central office equipment associated with the remote testing system.

3.2.3 Differences between host and RSS LTD line testing

Since the two testing options (RLT or MLAP) require different trunk circuit hardware at the central office, the incoming call from the LTD may appear on either type of trunk. This requires the LTD operator to know the testing option associated with a particular RSS, and it requires the host No. 2B ESS to verify that the incoming LTD trunk type agrees with the testing option of that RSS. This latter confirmation is performed when the line's directory number is translated into an RSS identity.

The Local Test Cabinet (LTC), normally located near the distributing frame in the central office, is not equipped with the remote testing system. Thus, the LTC is incompatible with RLT-equipped RSSs and can test RSS lines only if that RSS is equipped with the MLAP option.

Another difference between LTD/LTC testing of host versus RSS lines concerns the action taken if the line is found busy when the initial test connection is made. When a host line is being tested, a suitably marked incoming LTD trunk can be connected to the busy line by means of the no-test facility of the No. 2B ESS switching network. If the host line becomes idle during the LTD connection interval, the line is marked busy (again) and the LTD is reconnected using a normal network path. In the analogous RSS case, the LTD is bridged onto the existing line connection in the RSS network. Since the software line status may not be changed, a camp-on request is registered at the RSS. When the line becomes idle, the pending LTD request causes the line to be rebusied, but no path reconfiguration is required.

One of the LTD functions is to verify whether a given line can originate. This is done on host lines by requesting the line attending element (ferrod) to be reconnected to the customer loop and then placing a resistive bridge across the loop. When the off-hook condition is sensed, dial tone is returned to the tester via the LTD trunk circuit, as shown in Fig. 15. Such operation is not possible with RLT testing of an RSS line since returning dial tone through the LTD trunk would open the LTD-RLT path and prevent further communication between them. Instead, when a line origination test is to be performed on an RSS line, that line is connected to a digit receiver in the host (shown in Fig. 16). The digit receiver then performs the functions of: (1)

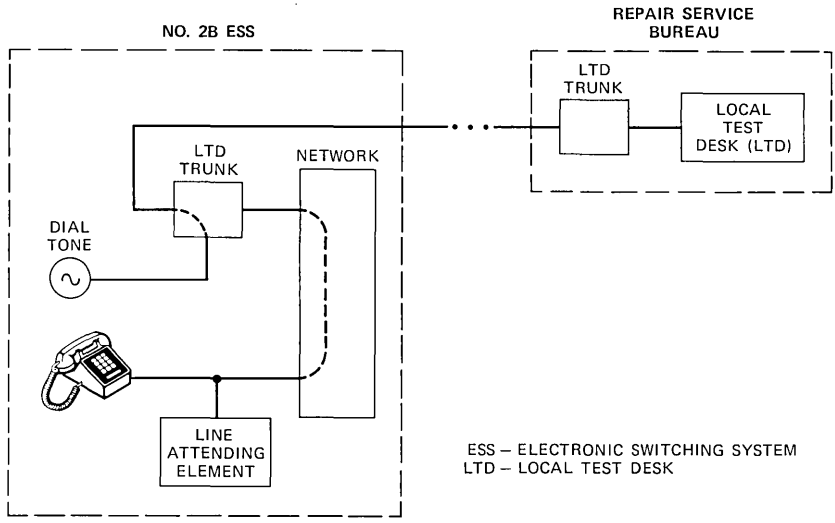


Fig. 15—Local test desk line origination test for host lines.

detecting supervisory changes on the line, and (2) returning dial tone when the off-hook signal is detected. Dial tone attachment and removal can then be monitored by the LTD, which is connected to the customer's line.

3.2.4 Station ringer testing

The station ringer test is under the control of the customer station and permits the verification of:

1. Station dialing capability
2. Party-identifying ground (in the off-hook state)
3. On-hook leakage, and
4. Ringing code and ring trip.

Since the sequence of actions required to perform these tests is well known to craft accustomed to testing host lines, a basic requirement for the service was that the craft interface remain unchanged for RSS lines.

The procedure, and its implementation for RSS lines, is diagrammed in Fig. 17. The test sequence begins by dialing a special code, normally a unique NNX, followed by the last four digits of the line's directory number. The RSS line is then connected to the host's station ringer test circuit via the same channel that had previously been connected to the customer digit receiver. The first step, the off-hook resistance test, is initiated with a switchhook flash by the tested line. A data-link order sent to the RSS causes the resistance test to be made by a

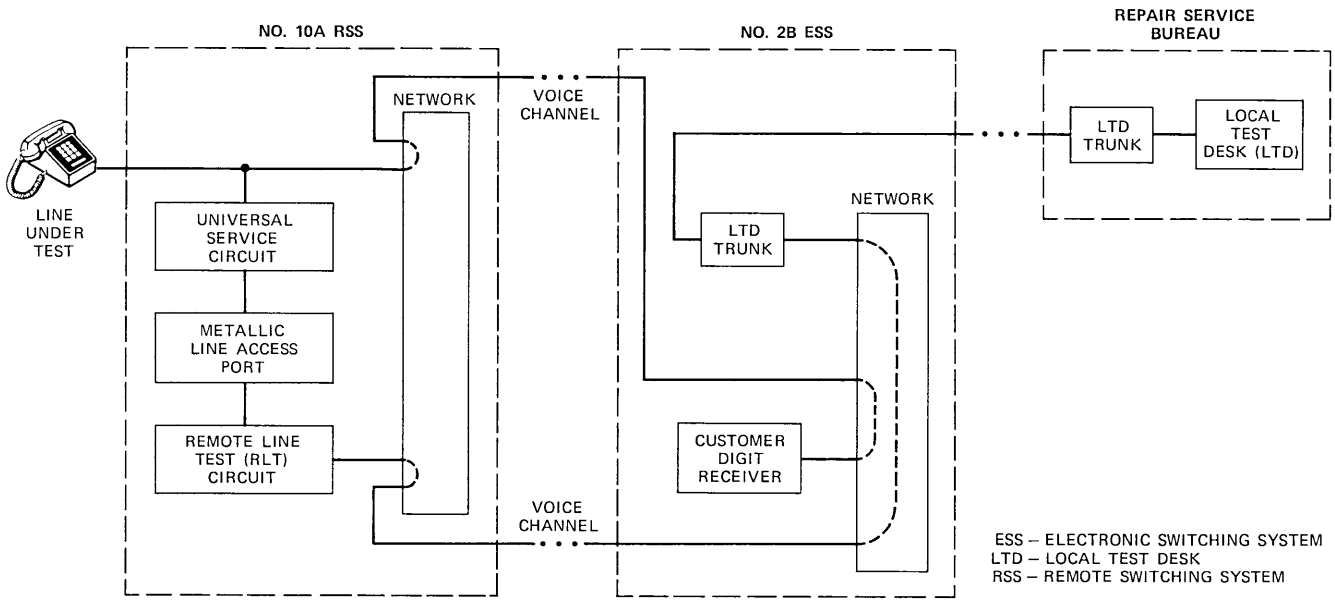


Fig. 16—Local test desk line origination test for RSS lines.

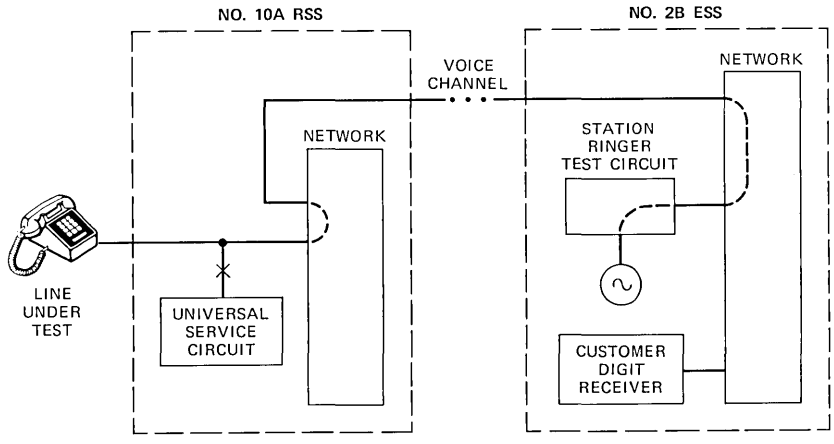


Fig. 17—Station ringer testing for RSS lines.

USC. The results of this test are then indicated with a steady vs. interrupted tone returned by the station ringer test circuit.

The second test on a (noncoin) customer line is an on-hook leakage test and is initiated by on-hook supervision. Again, the USC at the RSS is directed to perform the resistance test; its results are returned to the host via data-link message. If the resulting leakage measurement is acceptable, terminating ringing is applied to the line by the USC. Ringing indicates the result of the leakage test as well as confirming that the station set ringer is functional and that the correct ringing code is applied. The final test, ring trip, is made if the station set is taken off-hook within the next 3 minutes. As in the previous instances, since the ringing is applied by the RSS's USC, the tripping of ringing must be reported to the host via a data message.

At this point, the station ringer test circuit is connected to the line under test. A station set on-hook would idle all connected resources or, if a repeat test is desired, a switchhook flash causes the off-hook resistance test to be made, starting another test cycle.

3.3 DTMF receiver testing

As we mentioned previously, the RSS is capable of "stand-alone" operation should its interface with the host be lost. This feature therefore required that DTMF receivers be equipped in the RSS so that customers utilizing *Touch-Tone** dialing may be served. To

* Trademark of AT&T.

achieve design economies in the RSS, a DTMF test circuit (like that in the host) was not provided. The alternative to manual routine testing was to provide an automatic diagnostic using the host's DTMF test circuit.

Such operation is possible because the stand-alone circuits (including the DTMF receivers) have access to the RSS voice channels via a link "multiple" arrangement shown in Fig. 18. This network design, originally intended for mutually exclusive connections of RSS lines to either channels or stand-alone circuits, permits a voice channel connection to a DTMF receiver without use of the RSS network junctors. The resulting configuration during the execution of the diagnostic involves the host's DTMF receiver test circuit, as shown in Fig. 19.

Maintenance software is furnished to diagnose one or more receivers on a demand or automatic basis. For the automatically initiated case, diagnostics of the RSS receivers are carried out one RSS at a time following the corresponding diagnostics for host digit receivers.

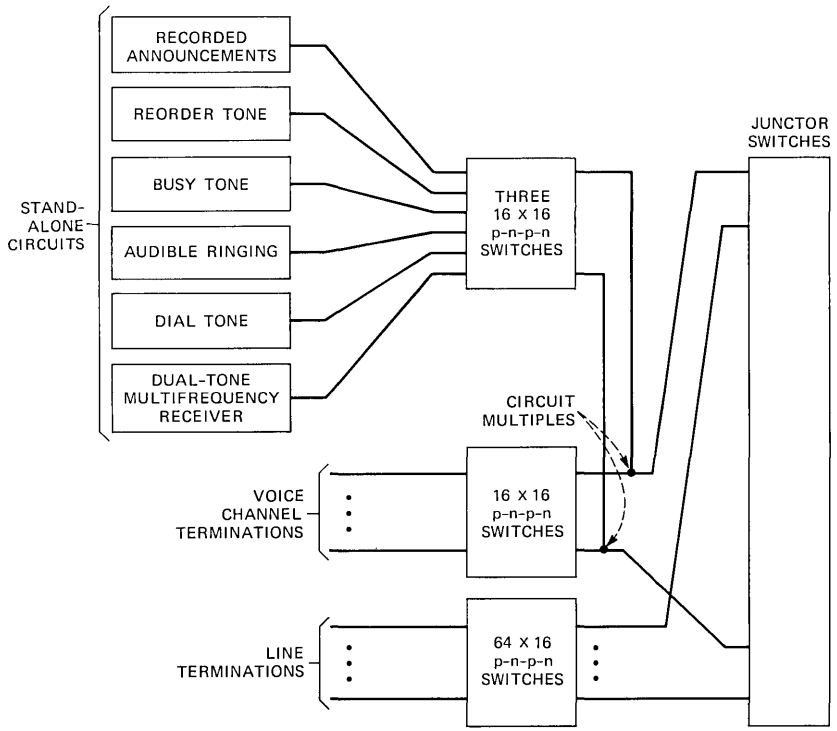


Fig. 18—Stand-alone circuit network connections.

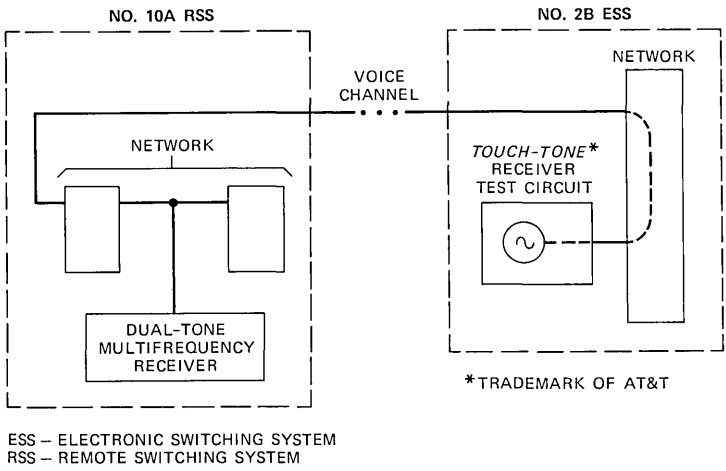


Fig. 19—*Touch-Tone* receiver diagnostic connections.

IV. REAL-TIME PERFORMANCE

This section discusses two aspects of the RSS feature: the real time required to process an RSS call and the effect of RSS on the call capacity of an office.

The many components of an RSS call were determined by performing extensive measurements in the system laboratory. The results of this study indicated that an ESS line-to-RSS line call requires about 33 percent more processor real time than a host line-to-line call. Similarly, an RSS line-to-ESS line call requires about 49 percent more real time and an intra-RSS call (reswitched-down) requires about 113 percent more processor real time than a host line-to-line call.

Because of overhead associated with each RSS hosted by a No. 2B ESS and real-time factors associated with each RSS call, the No. 2B ESS call capacity is reduced with an increasing proportion of RSS traffic. The amount of this reduction is a function of the number of RSSs hosted and the RSS traffic.

When this project was still in the initial planning stages, it was evident that providing RSS capabilities would affect the host office capacity. Throughout the entire development of the feature, considerable emphasis was placed on performance issues. This included such items as: planning effective communication strategies between the host and SPUC, minimizing RSS-related work in non-RSS segments of code, and optimizing and fine tuning frequently executed functions. The result of these activities is a relatively small penalty in host capacity due to RSS.

V. ACKNOWLEDGMENTS

The authors wish to acknowledge the outstanding efforts of the many individuals at Bell Telephone Laboratories, American Telephone and Telgraph, and Western Electric who have contributed to this work.

AUTHORS

Donald W. Brown, B.S.E.E., 1962, University of Louisville; M.S.E.E., 1964, New York University; Bell Laboratories, 1962—. Mr. Brown has been engaged in the system design of proposed Signal Transfer Points for the Common Channel Interoffice Signaling network. Since 1976, he has worked on system and program design for the Remote Switching System. He was past Supervisor of the RSS Remote Terminal Maintenance Group and is now Supervisor of the Cell Maintenance Group for the Advanced Mobile Phone Service system.

John J. Driscoll, B.S.E.E., 1964, Clarkson College of Technology; M.S.E.E., 1966, Stevens Institute of Technology; Bell Laboratories, 1964—. Mr. Driscoll has been engaged in various aspects of electronic switching system development. Initially, he worked on maintenance and call-processing software development for No. 2 ESS, and later he worked on exploratory studies and call-processing software development for the Advanced Mobile Phone Service. He was involved with call-processing and administrative software development on the No. 1 ESS and the No. 2B ESS RSS systems. Presently, he is working on No. 5 ESS performance evaluation and capacity improvement.

Frederick M. Lax, B.S.E.E., 1974, University of Notre Dame; M.S.E.E., 1975, Massachusetts Institute of Technology; Bell Laboratories, 1974—. From 1975 to 1979, Mr. Lax was engaged in software development for No. 1/1A ESS call-processing and No. 10A Remote Switching System (RSS) feature development. In 1979, he was named Supervisor of the RSS First Application/General Availability Features Group and, subsequently, Supervisor of the No. 5 ESS Audit Systems Group. He is presently Head of the No. 5 ESS Software Recovery Department.

Michael W. Saad, B.S.E.E., 1964, M.S.E.E., 1965, University of Illinois; Bell Laboratories, 1966—. M. Saad first worked in the maintenance software area for the Automatic Voice Network (AUTOVON) system. His No. 4 ESS experience includes call-processing design, peripheral error-analysis strategies and design, peripheral-unit-configuration program design, and No. 4 ESS translations for international switching. Mr. Saad was supervising the 2B ESS RSS Call-Processing Group and is currently supervising the Administrative Services Group for No. 5 ESS.

J. G. Whitemyer, B.S.E.E., 1963, University of Akron; M.S.E.E., 1965, Ohio State University; Bell Laboratories, 1963—. Mr. Whitemyer joined Bell Laboratories in 1963 and worked in the Local Crossbar Switching Laboratory where he was involved in the development of Custom Calling Services and *Picturephone** Service on the No. 5 Crossbar System. He was promoted to

* Registered service mark of AT&T.

Supervisor in 1970 and, shortly thereafter, took responsibility for the system design of No. 3 ESS. Subsequent assignments included responsibility for circuit and microcode design of the 3A Central Control, field support for the No. 3 ESS and No. 2B ESS, and software design for the No. 2B ESS. He is presently Supervisor of the 2BE4 Project Management and Software Group. Mr. Whitemyer holds patents relating to the implementation of telephone switching features and telemetry.

THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 6, July–August 1983
Printed in U.S.A.

Modernization of the Suburban ESS:

Billing and Measurements Modernization

By J. P. LODWIG* and D. A. WARD*

(Manuscript received May 28, 1982)

With the introduction of the 2B Extended Feature Generic #3 (2BE3), No. 2B Electronic Switching System has modernized its arrangements for billing and traffic measurements. High-speed, synchronous data links are used to teleprocess billing information to the No. 1A Automatic Message Accounting Recording Center and to forward traffic data to the No. 1A Engineering and Administrative Data Acquisition System Center. This article describes the implementation of these features in the 2BE3 generic and how these new interfaces have resulted in more precise and more reliable data.

I. INTRODUCTION

The billing and traffic measurements capabilities available in the No. 2B Electronic Switching System (ESS)† today are the culmination of an evolutionary process spanning more than a decade. During that period, the ESS environment was changing at a very rapid pace, and as a result the billing and traffic measurements processes were continually being upgraded to meet the demands for each generic. Repre-

* Bell Laboratories.

† Acronyms and abbreviations are listed at the back of this issue of the *Journal.*

senting the latest capabilities for billing and traffic measurements in the No. 2B ESS are the No. 1A Automatic Message Accounting Recording Center (AMARC) Interface feature and the No. 1A Engineering and Administrative Data Acquisition System (EADAS) feature, respectively. Both the AMARC and EADAS features require near-real-time data collection at the No. 2B ESS. The data are then passed to I/O message buffers for subsequent transmission to an AMARC or EADAS data center. The AMARC formats the billing information for a Revenue Accounting Office (RAO) to use in preparing customer bills, while EADAS provides real-time surveillance and disperses traffic data to various Operation Support Systems (OSSs) for use in maintaining the switching system and planning future growth. Data to the AMARC and EADAS systems are transmitted via synchronous, 2.4-kb data links using the BX.25 protocol. The AMARC feature also uses 4.8-kb data links for those offices presenting too large a load for the 2.4-kb link.

1.1 No. 1A AMARC

Although several hardware and software versions of an AMARC exist today, the No. 2B ESS interfaces only with the No. 1A AMARC equipped with the 1AAM4 generic. The AMARC provides service to various types of switching systems today with several different interfaces giving flexible format, protocol, and data-link speeds between the switching systems offices and AMARC. The AMARC combines the data received from the switching systems into a format acceptable by an RAO. The data are then recorded on magnetic tape, which is sent to the RAO for processing into telephone bills for the customers. Figure 1 represents the interface of a No. 2B ESS with an AMARC.

1.2 EADAS

An EADAS data center (with the 1AED4 generic) connected to the No. 2B ESS (with the 2BE3 generic) receives traffic data and plant data from the No. 2B ESS. The traffic data are load measurements such as peg and usage counts for office totals, and peg, usage, and overflow counts for trunk and service circuit groups. The plant data are measurements that indicate the health of the system. Examples are control unit and peripheral unit diagnostic all-tests-passes and faults. The traffic and plant data sent to EADAS and processed by the EADAS data center are then used by other associated OSSs. The Network Operations Report Generator (NORGEN), part of EADAS, accesses the processed EADAS data directly for its near-real-time reports. Downstream OSSs process EADAS data after they have been sent to them via magnetic tape. Two such downstream OSSs to access the data via the Traffic Data Administration System (TDAS) are

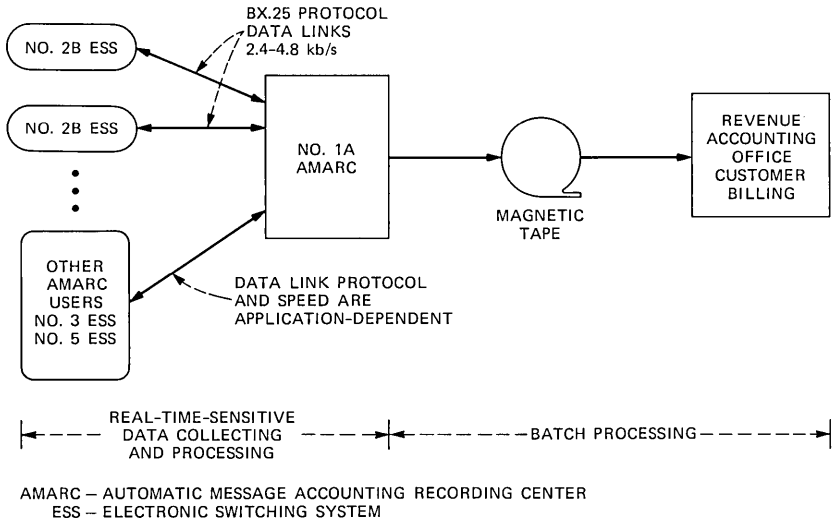


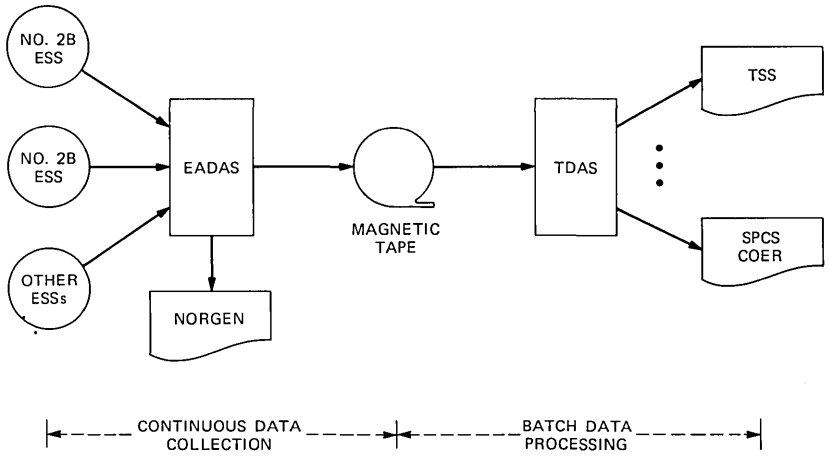
Fig. 1—Interface of a No. 2B ESS with an AMARC.

SPCS COER (Stored Program Control Systems Central Office Equipment Reports) and TSS (Trunk Servicing System). The reports generated by SPCS COER and TSS are needed by network administrators to monitor switching system services, measure utilization, and calculate capacity of the switches, and by trunk administrators to compute trunk group traffic load and current trunk requirements as part of their day-to-day jobs. The capability of providing timely data to SPCS COER and TSS is advantageous (see Fig. 2 for layout). The OSSs can now develop additional needed reports (e.g., summarizing centrex group counts). Also, with the addition of RSS host capability in 2BE3, the traffic data for an RSS is collected by the No. 2B ESS. Extreme Value Engineering (EVE) techniques are used for the engineering of an RSS. The EVE selection is done by EADAS and the engineering calculations are done by SPCS COER.

II. No. 2B ESS HISTORY AND LIMITS

2.1 Billing data collection

Prior to the installation of the 2BE3 generic, the primary method of billing calls was through Local Automatic Message Accounting (LAMA). Other methods used to a lesser degree were Centralized Automatic Message Accounting (CAMA) and in some instances message registers. The choice of billing arrangement was based on applicable tariffs (flat rate or local measured service) and resulting call volumes. Flat rate areas used message registers or CAMA while local measured service areas used LAMA.



COER – CENTRAL OFFICE EQUIPMENT REPORTS
EADAS – ENGINEERING AND ADMINISTRATIVE DATA ACQUISITION SYSTEM
NORGEN – NETWORK OPERATIONS REPORT GENERATOR
SPCS – STORED PROGRAM CONTROL SYSTEMS
TDAS – TRAFFIC DATA ADMINISTRATION SYSTEM
TSS – TRUNK SERVICING SYSTEM

Fig. 2—No. 2B ESS and Operation Support Systems.

The LAMA version of No. 2B ESS billing provides the most complete and versatile billing process. However, in some cases daily collection of LAMA tapes could be costly. One possible solution to this first problem was to enhance the software and upgrade the tape recorder to accommodate full measured service billing. However, the collection of the tapes on a daily basis still presents an unattractive cost factor.

2.2 Traffic data collection

In a No. 2B ESS office, each teletypewriter (TTY) will serve as a primary output device for one function. A 110-baud data link could be connected from the traffic TTY controller circuit to an EADAS monitored interface. Once it is output to any device, the data no longer exist in the corresponding register, and subsequent output to any other device is not possible. In addition, a large office could not transmit all of the data to EADAS because of the limited data-link speed. Thus, only a subset of the total data was sent (i.e., one or two hours worth each day, assigning different data to the schedules). Another problem was the data skew associated with these output processes. The TTY print rate was 10 characters per second. At this rate it took a minimum of 5 minutes to print a traffic schedule of 600 registers. (Most No. 2B offices have more than 600 registers on a schedule and some require

15 minutes print time for even larger schedules.) The last line of data on a printed schedule is therefore skewed from the first line by the print time. To solve the need for all the data all the time and for accurate data (i.e., prevent data skew) holding registers are added to the 2BE3 generic.

III. SOLUTION AND NEW PROBLEMS

The inclusion of AMARC and EADAS in the 2BE3 generic provides the increased billing capacity required by the switch and improved traffic information for engineering the switch. This does not preclude the use of billing arrangements available prior to 2BE3 (e.g., LAMA), where increased billing capacity is not required. Also, most of the traffic information capabilities available prior to 2BE3 have been retained. However, as with any new approach, new problems appeared for both AMARC and EADAS. In the early planning phase, one of the more basic problems was establishing interface requirements acceptable to Bell Laboratories systems engineering, AMARC development, EADAS development, and the No. 2B ESS development organization. The selection of a data-link protocol that could be used for both EADAS and AMARC was a highly desirable consideration. Also, during this same time frame, No. 5 ESS committed to AMARC and EADAS for its first application. This further complicated the situation in that now there was another system with a dissimilar architecture and software environment that could affect the development schedule and interface requirements.

IV. NEW PROTOCOL

Selection of the communications protocol was a major consideration in establishing AMARC and EADAS requirements. A protocol represents a formal agreement on the exchange of information between two or more entities. It provides a multilayered set of rules that govern the interconnecting electrical signals (level one), packetized data (level two), complete message (level 3), and user application data transfer procedures (level 4 and above). Several protocols were considered, e.g., DDCMP, BYSYNC, X.25, etc., but only X.25 had the recommendation of the Comite Consultatif International Telephonique et Telegraphique (CCITT). Eventually a subset of X.25, now referred to as BX.25, was proposed and accepted as the AMARC feature protocol. Subsequent investigation of this proposal revealed that the proposed BX.25 protocol was also suitable for the EADAS feature. A more detailed presentation of this capability in the No. 2B ESS 2BE3 generic is provided in this issue of the *Journal* in an article entited "Adding Data Links to an Existing ESS."

V. INTEGRATION OF AMARC

The AMARC feature encompassed many areas of call processing as well as data integrity and transmittal. Some areas were related to AMARC alone, while others were a consequence of a retrofit constraint that required simultaneous recording by AMARC and LAMA. Rather than attempting to cover the entire scope of AMARC, we will discuss only the basic operational capability and selected concerns.

5.1 Basic operation

The AMARC feature collects billing data in real time in the No. 2B ESS and then transmits this data using either a dedicated 2.4- or 4.8-kb/s full-duplex transmission facility or an automatic-dialed backup data-link facility of the same transmission speed in case of primary link failure. The No. 2B ESS/AMARC interface uses a double-entry billing system consisting of an initial/answer entry and a disconnect entry. LAMA recording, however, uses a triple-entry billing system consisting of initial, answer, and disconnect entries. Data for each LAMA entry are collected in real time and loaded into a dedicated LAMA buffer prior to being directed to the tape recorder via I/O control at interrupt level. The use of special AMA registers allows the basic billing structure to remain as it had been for LAMA and still achieve the double-entry billing required by AMARC. Each AMA register is linked with a specific call by direct extension of the Transient Call Record (TCR) memory already designed into the No. 2B ESS architecture. These TCR extensions are then referred to as TCRX registers. The initial billing data are collected and placed into the TCRX registers until answer time and then, upon verification of Minimum Chargeable Duration (MCD), the data are moved into an AMARC data buffer. At disconnect, the data are again collected in the TCRX. When the entry is assembled, it is then loaded into the AMARC data buffer.

5.2 Entry association

An obvious requirement for assembling the double entry into one call record by AMARC is the need to associate the initial/answer entry with the corresponding disconnect entry. The method provided by the No. 2B ESS is to map an equipment number (the physical location of a call on the No. 2B ESS or RSS network) into a 15-bit Virtual Equipment Number (VEN) and record the VEN as the low-order 16 bits of a 24-bit Call Assembly Index (CAI) that accompanies each billing entry. The CAI high 8 bits are zeroes except for the case of call-forwarded calls. The special use of the high 8 CAI bits for call-forwarded calls will be described later in this article.

The CAI counterpart in LAMA recording is the Call Identity Index

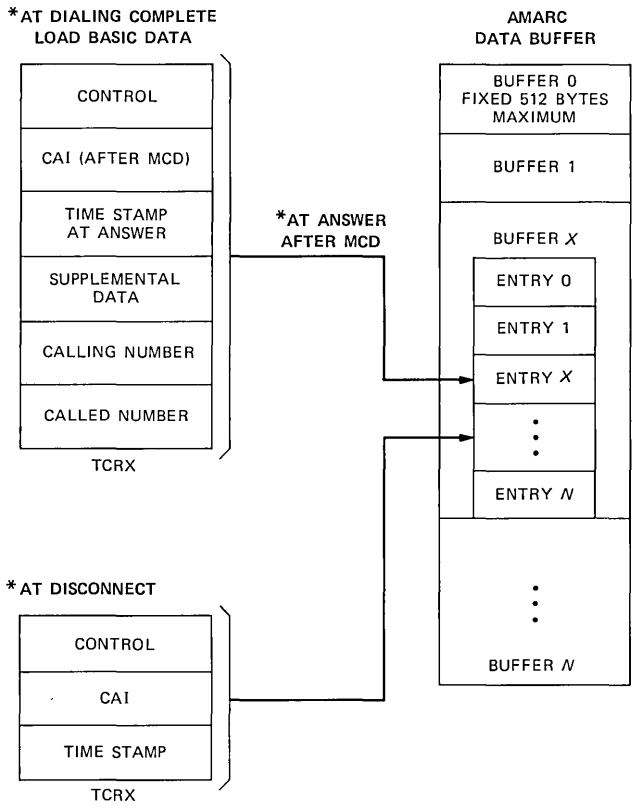


Fig. 3—Basic billing information flow between internal AMARC registers and buffers.

(CII). LAMA recording guaranteed that CIIs appearing on the tape would be unique for the duration of a call by maintaining a table of the active CIIs derived using a time-consuming hashing scheme based upon the VEN. Since the method of call sequencing associated with the CAI no longer requires the ESS to provide a hashing algorithm, significant real-time savings are achieved for the No. 2B ESS call-processing execution environment. Figure 3 represents the basic billing information flow between internal AMARC registers and buffers.

5.3 AMARC data buffer

The No. 2B ESS AMARC data buffer is an engineerable area of read/write memory. The maximum size is designed to provide up to two minutes of billing data storage during a data-link failure. Auto-

matic recovery to the dialed backup data link can be obtained within this two-minute period. The buffer is segmented into message size entities of 512 bytes each. When a buffer segment has insufficient room for the next data entry, or a fixed time limit has expired, the data are either sent immediately or queued for later transmission through the I/O program. The decision depends on available buffer space on the Serial Peripheral Unit Controller for Data Links (SPUC/DLs). Further information on this subject can be found in the article "Adding Data Links to an Existing ESS," also in this issue of the *Journal.*

5.4 Real-time benefit

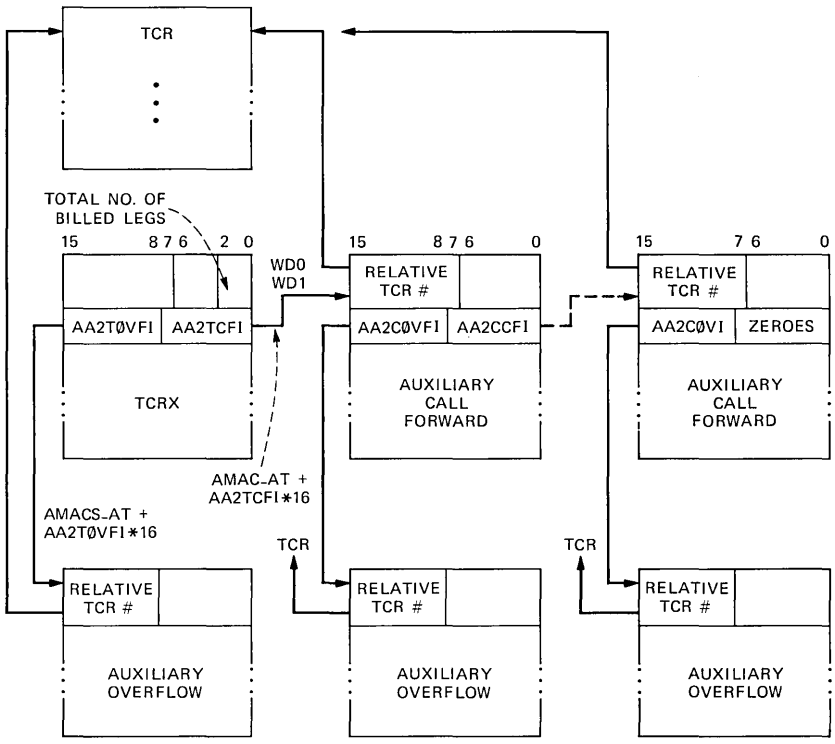
The I/O process that moves the data buffer to a SPUC/DL is accomplished at a significant real-time savings compared with the I/O interrupt process for LAMA. Whereas LAMA had a data acceptance limitation (primarily related to hardware design) of three bytes of data within a 20-millisecond interval, the hardware structure of the SPUC/DL allows software definition of the maximum number of data bytes in a message. The I/O design in 2BE3 is flexible but currently allows 256 bytes of data to be sent to the SPUC/DL during one 100-millisecond cycle of the operating systems main loop. Thus, the repetitive I/O overhead associated with small segments of information pertaining to each LAMA recorded call is very significant compared to the AMARC overhead for sending 256 bytes of data for 15 to 20 calls. The overall real-time savings due to the AMARC billing feature, including I/O changes, was 2.8 milliseconds per call (3.6-ms AMARC versus 6.4-ms LAMA). This has the effect of increasing the overall call-billing capacity of the No. 2B ESS when full measured service is offered.

5.5 Auxiliary registers

For some cases where the data required for a specific call type cannot be contained in a single TCRX register, auxiliary registers are provided to hold the overflow data and are referred to as auxiliary overflow registers. The auxiliary registers are unique to the AMARC feature, while the TCRX registers may be used for other features not related to call processing. The auxiliary registers are associated with the call by placing an index pointer into the TCRX used for the call. Figure 4 shows the connectivity of auxiliary overflow registers plus an extension into auxiliary call-forwarding registers.

5.6 Multiple calls associated with one CAI

Each leg of a call-forwarded call requires separate billing to each of the parties involved with the call. Since separate billing is required,



TCR – TRANSIENT CALL RECORD
TCRX – TRANSIENT CALL RECORD EXTENSION REGISTER

Fig. 4—AMARC message register interrelation.

the simplest method of administration is to provide a separate initial/answer entry followed by a corresponding disconnect entry for each leg of the call. This requires a different CAI for each billed leg and thus presented a problem in that the only VEN available at disconnect is the originating party VEN. This occurred since the No. 2B ESS call-processing structure prior to the 2BE3 generic did not provide a record of the intervening call legs once the call was made stable. The LAMA billing relied on the CII record table to provide the correct billing entry association. In the 2BE3 generic a record of the number of billed legs of the call is kept in a Stable Information Entry (SIE) during the stable state of the call. The SIE itself is accessed based upon the originating party VEN and codes defining the specific use for that SIE.

Correct billing entry association for AMARC call-forwarded billing records is achieved by prefixing an incremental count to the CAI at answer time for each call-forwarded billing entry except for the first

billed leg of the call. The CAI can then be expressed as the CAI = (N-1)*2**16 + VEN, where VEN represents the 16 low-order bits of the CAI and N equals the billed leg number (1–3) for call-forwarded calls. At disconnect, a billing entry is made for each billed leg of the call by once again prefixing a count to the CAI for each additional billed leg of a call exceeding the first leg. The correct number of disconnect entries is determined by the data available in the SIE.

The initial billing information for call forwarding is collected in the associated TCRX and in supplemental auxiliary registers as required. As in the case of auxiliary overflow registers, call-forward registers are associated with the call by loading an index point to the current register into the register servicing the previous leg of the call. An audit capability is provided by placing the TCR number in each register. Thus, random audits may test a register for activity and, by going to the specific TCR, test for the expected connectivity back to the auxiliary registers originally interrogated. This connectivity is illustrated in Fig. 4.

5.7 Retrofit considerations

Several items of concern exist for the case of an AMARC retrofit into an office previously served by LAMA. One concern relating to the LAMA/AMARC simultaneous recording requirement during retrofits is the administration of call types, e.g., measured service, toll, etc. LAMA entry codes (equivalent to AMARC call types) relating to various calling conditions do not map directly with AMARC call types. This required careful preparation of administrative documentation to allow in some cases a temporary association of call-type categories for LAMA and AMARC. A significant consequence of the call-type association is the ability of the telephone operating companies to validate system operation by comparing the billing records of LAMA and AMARC during the retrofit mode of operation. Another consideration for the retrofit case is the recovery strategy. Since LAMA was the prime billing medium prior to retrofit, all calls billed during the retrofit are obtained from LAMA data tapes; thus all failure modes default to promoting successful LAMA billing.

VI. INTEGRATION OF EADAS

The EADAS feature modified the traffic and plant-collecting and printing routines. With the BX.25 protocol as a common base, developing the interface requirements became a matter of resolving application-level interfaces between No. 2B ESS, No. 5 ESS, and No. 1A EADAS. There are interface differences between the No. 2B ESS to No. 1A EADAS and the No. 5 ESS to No. 1A EADAS due to the

different architecture structures of No. 2B ESS/generic 2BE3 and No. 5 ESS. Having the No. 2B ESS conform to the identical interface between No. 5 ESS and No. 1A EADAS would cause a real-time penalty to No. 2B ESS. The resulting operational interface between the No. 2B ESS and No. 1A EADAS is described next.

6.1 Basic operation

The high-speed EADAS interface feature provides for the collection and transmittal of both traffic and plant measurement data to No. 1A EADAS. With this implementation the traffic and plant measurement data are collected on two schedules—a 30-minute schedule and a 24-hour schedule. The schedules start collecting 30 seconds before the clock 00 minute and 30 minutes and finish by one minute after 00 or 30. The data are collected and stored in holding registers within 90 seconds, resulting in a maximum of 90 seconds of data skew. The data for most offices will be collected within 30 seconds. The resulting data skew of 30 to 90 seconds is significantly more accurate than the previous skew of 5 to 15 minutes. Schedules are not transmitted until a poll is received from No. 1A EADAS requesting a schedule. The No. 2B ESS and No. 5 ESS communicate with the 1AED4 generic of No. 1A EADAS using the BX.25 protocol.

The interface between the No. 2B ESS and the No. 1A EADAS is a poll-and-answer mechanism. The No. 1A EADAS polls for four types of data:

1. Time of day
2. 30-minute data
3. 24-hour data
4. Record base (working member) data.

The No. 2B ESS answers with the:

1. Time of day
2. Data from the H (specified busy hour), C (specified nonbusy hour), W (specified weekly), and PLT (specified plant) schedules
3. D (daily or division of revenue) schedule
4. Working member count of trunk and service circuits for each poll.

The No. 1A EADAS also sends a message when it is planning to discontinue operation temporarily for such things as maintenance. The No. 2B ESS accepts the message and prints out a message to the TTY to inform the craft. The No. 2B ESS can send error response messages to No. 1A EADAS: (1) when the data for a message have been overwritten because current polling from the No. 1A EADAS is still continuing when the next collection is started, or (2) if a poll comes in when the 2B is collecting data for a message.

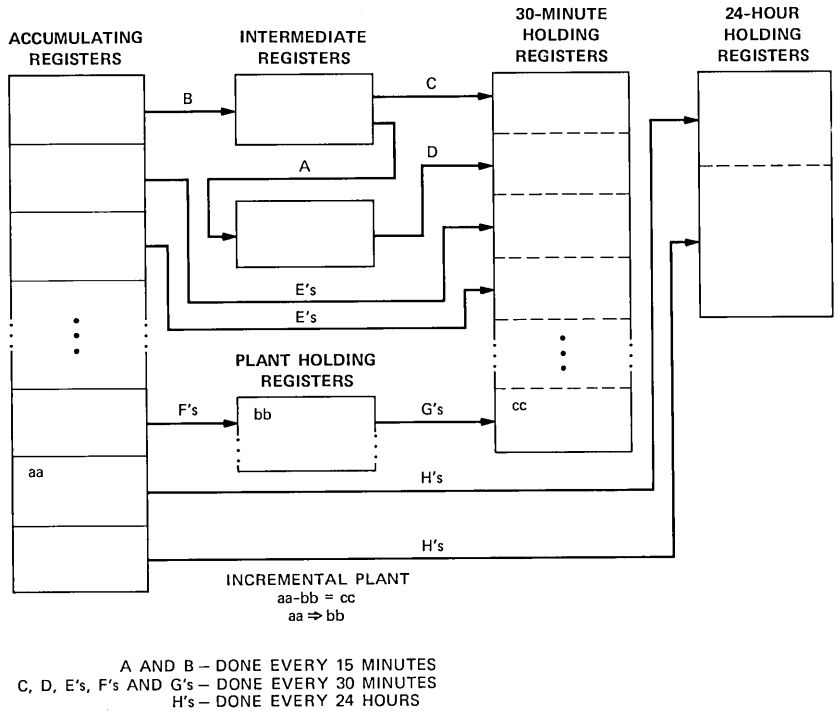


Fig. 5—Data collection.

6.2 Data collection

Data collection is started every quarter hour to collect the intermediate counts and every half hour to collect the 30-minute counts. Also, the 24-hour counts are collected at midnight before the 30-minute counts. In Fig. 5 the lines A through H match the description of each type of count. The quarter-hour counts, load service measurements, are moved to the intermediate registers (A and B). The intermediate registers contain two sets of these registers, the previous quarter hour's and the current quarter hour's. The two sets of counts are kept until the half hour when both sets are moved to the 30-minute holding registers (C and D). Also, every half hour the counts such as office totals, service circuit, trunk circuit, multiline hunt, and simulated groups, junctor usage, centrex, and network usage are moved to the 30-minute holding registers (Es). Then the half-hour increment of the plant counts are moved to the 30-minute holding registers (Fs and Gs). This is done by subtracting from the current plant accumulating counts the intermediate plant holding registers, storing the difference in the 30-minute holding registers, and moving the accumulating plant

counts to the intermediate plant holding registers. The 24-hour counts, division of revenue, are moved to the 24-hour holding registers (Hs).

6.3 Override No. 1A EADAS

If necessary, the 2BE3 generic can revert to the standard H, C, W, and D schedules through human intervention. When an office has EADAS active, the 30-minute data are from all the data that could have been on the H, C, and W schedules and the 24-hour data are from the D schedule. An office could revert to the standard schedule output if desired or if the No. 1A EADAS center would go off-line for an extended period of time. The No. 2B ESS office would then provide the data on hard copy and paper tape punch. In the No. 2B ESS, the standard schedules are printed automatically under the control of the Traffic Work Table (TWT) from the holding registers. This arrangement (non-EADAS) does not support RSS EVE engineering needs for SPCS COER. This mechanism is overridden when EADAS is active and it can be reenabled with a single TTY input message to inhibit EADAS.

6.4 Interface differences

Because of the differences in No. 2B and No. 5 ESS architecture, the interface with No. 1A EADAS does differ on three points:

1. The No. 2B ESS keeps time differently than the No. 5 ESS.
2. The No. 2B ESS does not indicate overflow of registers through the special register values as does No. 5 ESS. The values 65,526 through 65,535 are used as special register values, i.e., for overflow, bad data, and unequipped register.
3. The 2B transmits two-byte data low byte first instead of the reverse.

Therefore, the No. 1A EADAS does handle these differences. For the No. 2B ESS to have followed the same interface as between No. 5 ESS and No. 1A EADAS on these three points would have cost the No. 2B ESS a real-time penalty.

VII. SUMMARY

The addition of the AMARC and EADAS features to the 2BE3 generic enhanced the No. 2B ESS capability. AMARC achieves the billing systems goals of reduced operating expense while providing full measured service billing capability for the No. 2B ESS. Also, owing to a redistribution of tasks and restructuring of the programs, AMARC allows higher call capacity in the No. 2B ESS compared with LAMA billing.

All traffic and plant data are sent to EADAS, which allows for the efficient integration of the local switching systems into Total Network

Operation Plan (TNOP). Also, owing to the addition of holding registers there is virtually no data skew. The EADAS feature is using a communication interface that could support network management messages and controls in a future generic.

VIII. ACKNOWLEDGMENTS

The authors extend their appreciation to all of the 2BE3 development team for their cooperation in providing the AMARC and EADAS capabilities for the No. 2B ESS. In particular, we wish to thank Iris Dowden and Eric Kampmeier for their contributions towards AMARC and Wendell Savino for his participation with EADAS.

AUTHORS

Deborah A. Ward, A.A.S. (Computer Technology), Purdue University Calumet Campus; Bell Laboratories, 1973—. After joining Bell Labs in 1973, Mrs. Ward worked on No. 3 ESS and No. 2B ESS projects. During this time, she was involved in the translations design and the program design, coding and testing of the translations access routines for No. 3 ESS. She also was involved in maintaining the administrative traffic program for No. 3 ESS. Her work on No. 2B ESS has also been in the administrative traffic area. Mrs. Ward has been a member of the No. 5 ESS Data Base Design Department at Bell Laboratories in Naperville, Illinois, and since early 1982 has been working on the No. 5 ESS project.

John Lodwig, B.S.E.E., 1971, Illinois Institute of Technology; Bell Laboratories, 1967—. Mr. Lodwig's initial work at Bell Laboratories involved circuit design with the No. 101 ESS, and later the 3A Common Controller, which has been used as a central processor for several electronic switching systems, e.g., No. 2B ESS, No. 3 ESS, etc. In 1976, he made the transition from circuit design to software design and became involved with automatic message accounting billing. It was during the period from 1976 through 1980 that he became involved with the Automatic Message Accounting Recording Center (AMARC) billing feature for the No. 2B ESS. More recently, Mr. Lodwig has been working with common channel interoffice signaling and domestic local planning for the No. 5 ESS.

ACRONYMS AND ABBREVIATIONS

2BE3	2B extended feature generic #3
3A CC	3A Processor Control Complex
ALIT	automatic line insulation testing
AMA	automatic message accounting
AMACS	automatic message accounting collection system
AMARC	automatic message accounting recording center
APC	AMARC protocol converter
CAI	call assembly index
CAMA	centralized automatic message accounting
CAROT	centralized automatic reporting on trunks
CCITT	Comite Consultatif International Telephonique et Telegraphique
COER	central office equipment reports
CII	call identity index
CRC	cyclic redundancy check
DDD	direct distance dialing
DL	data link
DLI	data-link interface
DMA	direct-memory access
DN	directory number
DTMF	dual-tone multifrequency
EADAS	Engineering and Administrative Data Acquisition System
EMF	electromotive force
ESS	electronic switching system
EVE	extreme value engineering
FCG	false cross or ground
FIFO	first-in/first-out
I/O	input/output
LAMA	local automatic message accounting
LAP	link access procedure
LAPB	link access procedure B
LTAB	line test access bus
LTC	local test cabinet
LTD	local test desk
MAB	metallic access bus
MCD	minimum chargeable duration
MF	multifrequency
MLAP	metallic line access port
MSF	multi-scan function
NORGEN	network operations report generator
OSS	operation support system

P(R)	receive sequence number
P(S)	send sequence number
PMJ	path memory for junctor
PMR	path memory remote
POB	peripheral order buffer
PUC	peripheral unit controller
RAM	random access memory
RAO	revenue accounting office
RLT	remote line test
ROB	remote order buffer
ROH	receiver off-hook
ROM	read-only memory
ROTL	remote office test line
RSS	remote switching system
RTS	remote testing system
SCC	switching control center
SIE	stable information entry
SPCS	Stored Program Control Systems
SPUC/DL	serial peripheral unit controller/data link
TCR	transient call register
TDAS	Traffic Data Administration System
TNOP	Total Network Operation Plan
TRG	tip-ring to ground
TRL	transistor resistor logic
TSS	Trunk Servicing System
TTP	trunk test panel
TTY	teletypewriter
TWT	traffic work table
USART	universal synchronous-asynchronous receiver-transmitter
USC	universal service circuit
VEN	virtual equipment number

PAPERS BY BELL LABORATORIES AUTHORS

COMPUTING/MATHEMATICS

David A. J., Meyer G. G. L., **Unstructured Mean Iterative Processes in Reflexive Banach-Spaces.** Siam J Con 21(1):140–152, 1983.

Gehani N. H., **High-Level Form Definition in Office Information Systems.** Computer J 26(1):52–59, 1983.

Odlyzko A. M., **Minima of Cosine Sums and Maxima of Polynomials on the Unit-Circle.** J Lond Math 26(Dec):412–420, 1982.

ENGINEERING

Alferness R. C., Buhl L. L., **Long-Wavelength Ti:$LiNbO_3$ Waveguide Electrooptic TE↔TM Converter.** Electr Lett 19(2):40–41, 1983.

Aspnes D. E., Theeten J. B., **(Fr) Dielectric Semiconductor Interfaces Analysis Using Spectroscopic Ellipsometry.** Act Electr 24(3):217–227, 1982.

Bronez T. P., Cadzow J. A., **An Algebraic Approach to Superresolution Array-Processing.** IEEE Aer El 19(1):123–133, 1983.

Candy J. C., Wooley B. A., **Precise Biasing of Analog-to-Digital Converters by Means of Auto-Zero Feedback.** IEEE J Soli 17(6):1220–1225, 1982.

Cooper J. A., Capasso F., Thornber K. K., **Semiconductor Structures for Repeated Velocity Overshoot.** Elec Dev L 3(12):407–408, 1982.

Fichtner W., Watts R. K., Fraser D. B., Johnston R. L., Sze S. M., **0.15 μm Channel-Length MOSFETs Fabricated Using E-Beam Lithography.** Elec Dev L 3(12):412–414, 1982.

Forrest S. R., Kohl P. A., Panock R., Dewinter J. C., Nahory R. E., Yanowski E., **A Long-Wavelength, Annular $In_{0.53}Ga_{0.47}As$ P-I-N Photodetector.** Elec Dev L 3(12):415–417, 1982.

Liu S., Nagel L. W., **Small-Signal MOSFET Models for Analog Circuit-Design.** IEEE J Soli 17(6):983–998, 1982.

McCaughan L., Murphy E. J., **Influence of Temperature and Initial Titanium Dimensions on Fiber-TI-$LiNbO_3$ Wave-Guide Insertion Loss at Lambda = 1.3 μm.** IEEE J Q El 19(2):131–136, 1983.

Perry A., **The Two-Level, Single-Period Network Layout Optimization Problem and Its Solution.** Infor 20(4):336–356, 1982.

Tabatabaiealavi K., Choudhur A. N., Alavi K., Vlcek J., Slater N. J., Fonstad C. G., Cho A. Y., **Ion-Implanted $In_{0.53}Ga_{0.47}As/In_{3.52}Al_{0.48}As$ Lateral pnp Transistors.** Elec Dev L 3(12):379–381, 1982.

Tillman F. A., Kuo W., Hwang C. L., Grosh D. L., **Bayesian Reliability and Availability—A Review (Review or Bibliog.).** IEEE Reliab 31(4):362–372, 1982.

Vanderziel J. P., Logan R. A., **Dispersion of the Group-Velocity Refractive-Index in GaAs Double Heterostructure Lasers.** IEEE J Q El 19(2):164–169, 1983.

Vanderziel J. P., Temkin H., Logan R. A., **Quaternary 1.5 μm (InGaAsP/InP) Buried Crescent Lasers With Separate Optical Confinement.** Electr Lett 19(3):113–115, 1983.

MANAGEMENT/ECONOMICS

Stiglitz J., Weiss A., **Alternative Approaches to Analyzing Markets With Asymmetric Information—Reply (Letter).** Am Econ Rev 73(1):246–249, 1983.

PHYSICAL SCIENCES

Aspnes D. E., Studna A. A., **Dielectric Functions and Optical-Parameters of Si, Ge, GaP, GaAs, GaSb, InP, InAs, and InSb from 1.5 to 6.0 eV (Review or Bibliog.).** Phys Rev B 27(2):985–1009, 1983.

Baraff G. A., Schluter M., Allan G., **Cluster-Extended Greens-Function for Electronic-Structure of Localized Defects in Solids.** Phys Rev B 27(2):1010–1016, 1983.

Bokor J., Bucksbaum P. H., Auston D. H., **Measurement of Picosecond Ultraviolet-Laser Pulse-Widths Using an Electrical Auto-correlator.** Appl Phys L 42(4):342–344, 1983.

Bondybey V. E., English J. H., **Electronic-Structure and Vibrational Frequency of Cr-2.** Chem P Lett 94(5):443–447, 1983.

Boyle N. G., Mcbrierty V. J., Douglass D. C., **A Study of the Behavior of Water in Nafion Membranes.** Macromolec 16(1):75–80, 1983.

Cais R. E., Sloane N. J. A., **A Statistical-Theory of Directional Isomerism in Polymer-Chains and its Application to Polyvinylidene Fluoride.** Polymer 24(2):179–187, 1983.

Carroll P. J., Lannin J. S., **Vibrational Properties of Crystalline Group-Vi Solids—Te, Se, S.** Phys Rev B 27(2):1028–1036, 1983.

Chang R. P. H., Darack S., **Plasma Enhanced Beam Deposition of Thin Dielectric Films.** Appl Phys L 42(3):272–274, 1983.

Chin A. K., Zydzik G., Singh S., Vanuitert L. G., Minneci G., **Al_2O_3 as an Anti-Reflection Coating for InP/InGaAsP LEDs.** J Vac Sci B 1(1):72–73, 1983.

Cieplak M., Banavar J. R., **Sensitivity to Boundary-Conditions of Ising Spin-Glasses.** Phys Rev B 27(1):293–296, 1983.

Denbroeder F. J. A., Klerk M., Vandenberg J. M., Hamm R. A., **A Comparative-Study of Diffusion Induced Grain-Boundary Migration, Recrystallization and Volume Diffusion During the Low-Temperature Diffusion of Al Into Cu and Au.** Act Metall 31(2):285–291, 1983.

Downey P. M., Auston D. H., Smith P. R., **Picosecond Correlation-Measurements of Indium-Phosphide Photoconductors.** Appl Phys L 42(3):215–217, 1983.

Dubois L. H., Nuzzo R. G., **Small-Molecule Chemisorption of $NiSi_2$—Implications for Heterogeneous Catalysis.** J Am Chem S 105(3):365–369, 1983.

Dutt B. V., Brasen D., **TEM Observation of Precipitates of Cd-Phosphides in Cd-Diffused InP—A Correlation With the Previously Proposed Diffusion-Models.** J Elchem So 130(1):207–214, 1983.

Elhanany U., Brennert G. F., Warren W. W., **Enhanced Paramagnetism and Spin Fluctuations in Expanded Liquid Cesium.** Phys Rev L 50(7):540–544, 1983.

Flamm D. L., Donnelly V. M., Ibbotson D. E., **Basic Chemistry and Mechanisms of Plasma-Etching.** J Vac Sci B 1(1):23–30, 1983.

Fleury P. A., Lyons K. B., Katiyar R. S., **Acoustic Anomalies in $Tb_2(MoO_4)_3$ and the "Missing" A_1 Optic Mode.** Phys Rev B 26(12):6397–6407, 1982.

Fuerst C. O., Fischer J. E., Axe J. D., Hastings J. B., McWhan D. B., **Pressure-Induced Staging Transitions in KC_8: Observation of a Fractional Stage.** Phys Rev L 50(5):357–360, 1983.

Ginsberg A. P., Osborne J. H., Sprinkle C. R., **Electronic-Structure and Bonding in the Disulfur and Diselenium Complexes $[M(X_2)(PH_3)_4]^+$(M = Rh, Ir; X = S, Se)** Inorg Chem 22(2):254–266, 1983.

Hockberger P., Connor J. A., **Intracellular Calcium Measurements with Arsenazo-III During Cyclic-AMP Injections Into Molluscan Neurons.** Science 219(4586):869–871, 1983.

Huggins H. A., Gurvitch M. **Magnetron Sputtering System Equipped With a Versatile Substrate Table.** J Vac Sci A 1(1):77–80, 1983.

Hwang J. C. M., Brennan T. M., Cho A. Y., **Initial Results of a High Throughput MBE System for Device Fabrication.** J Elchem So 130(2):493–496, 1983.

Jackson C. M., Zettl A., Gruner G., Disalvo F. J., **Frequency-Dependent Conductivity in $HfTe_5$ and $ZrTe_5$.** Sol St Comm 45(3):247–249, 1983.

Kash K., Wolff P. A., Bonner W. A., **Non-Linear Optical Studies of Picosecond Relaxation-Times of Electrons in n-GaAs and n-GaSb.** Appl Phys L 42(2):173–175, 1983.

Kevan S. D., **Evidence for a New Broadening Mechanism in Angle-Resolved Photoemission from Cu(111).** Phys Rev L 50(7):526–529, 1983.

Klauder J. R., **Wiener Measures for Quantum-Mechanical Path-Integrals.** Lect N Phys 173:245–247, 1982.

Kuk Y., Feldman L. C., Silverman P. J., **Transition from the Pseudomorphic State to the Nonregistered State in Epitaxial-Growth of Au on Pd(111).** Phys Rev L 50(7):511–514, 1983.

Kurkjian C. R., Paek U. C., **Single-Valued Strength of Perfect Silica Fibers.** Appl Phys L 42(3):251–253, 1983.

Lee T. P., Burrus C. A., Linke R. A., Nelson R. J., **Short-Cavity Single-Frequency InGaAsP Buried-Heterostructure Lasers.** Electr Lett 19(3):82–83, 1983.

Levin R. M., **The Step Coverage of Undoped and Phosphorus-Doped SiO_2 Glass-Films.** J Vac Sci B 1(1):54–61, 1983.

Levine B. F., Logan R. A., Tsang W. T., Bethea C. G., Merritt F. R., **Optically Integrated Coherently Coupled $Al_x Ga_{1-x}$ As Lasers.** Appl Phys L 42(4):339–341, 1983.

Liminga R., Abrahams S. C., Glass A. M., Kvick A., **Temperature-Dependence of the Nuclear Positions and Spontaneous Polarization in Pyroelectric $Ba(NO_2)_2 \cdot H_2O$.** Phys Rev B 26(12):6896–6900, 1982.

Marshall J. H., **The Nickel Metal-Catalyzed Decomposition of Aqueous Hydrophosphite Solutions.** J Elchem So 130(2):369–372, 1983.

Menezes S., Miller B., Bachmann K. J., **Electrodissolution and Passivation Phenomena in III-V Semiconducting Compounds.** J Vac Sci B 1(1):48–53, 1983.

Menezes S., Miller B., **Surface and Redox Reactions at GaAs in Various Electrolytes.** J Elchem So 130(2):517–523, 1983.

OBryan H. M., Thomson J., **$Ba_2Ti_9O_2O$ Phase-Equilibria.** J Am Ceram 66(1):66–68, 1983.

Petroff P. M., **Device Degradation and Recombination Enhanced Defect Processes in III-V Semiconductors.** Sem Insul 5(3–4):307–319, 1983.

Phillips J. C., **Can Glass Flow—Reply (Letter).** Phys Today 36(2):95, 1983.

Piancastelli M. N., et al., **Pyridine Chemisorption on Silicon and Germanium(111) Surfaces.** Sol St Comm 45(3):219–221, 1983.

Rousseau D. L., Ondrias M. R., Lamar G. N., Kong S. B., Smith K. M., **Resonance Raman-Spectra of the Heme in Leghemoglobin—Evidence for the Absence of Ruffling and the Influence of the Vinyl Groups.** J Biol Chem 258(3):1740–1746, 1983.

Schwartz G. P., Gualtier G. J., **Evaluation of Aluminum-GaAs Schottky Barriers Using Norde Modified Current-Voltage Analysis.** Appl Phys L 42(3):265–267, 1983.

Seliskar C. J., Heaven M., Leugers M. A., **The Free Jet Spectrum of the Toluene 2668-A Origin Band.** J Mol Spect 97(1):186–193, 1983.

Sermage B., Eichler H. J., Heritage J. P., Nelson R. J., Dutta N. K., **Photo-Excited Carrier Lifetime and Auger Recombination in 1.3-μ-M InGaAsP.** Appl Phys L 42(3):259–261, 1983.

Shank C. V., **Measurement of Ultrafast Phenomena in the Femtosecond Time Domain.** Science 219(4588):1027–1031, 1983.

Simpson A. M., Jericho M. H., Disalvo F. J., **Elasticity and Thermal-Expansion of 2H-$TaSe_2$ Between 4K and 130K** Sol St Comm 44(12):1543–1546, 1982.

Tam S., Hsu F. C., Ko P. K., Hu C., Muller R. S., **Hot-Electron Induced Excess Carriers in MOSFETs.** Elec Dev L 3(12):376–378, 1982.

Tersoff J., Bayer D., **Quantum Statistics for Distinguishable Particles.** Phys Rev L 50(8):553–554, 1983.

Vaidya S., Sinha A. K., Andrews J. M., **Electromigration Induced Shallow Junction Leakage with Al Poly-Si Metallization.** J Elchem So 130(2):496–501, 1983.

Williams J. H., et al., **Quantitative Geometric Characterization of Two-Dimensional Flaws via Liquid-Crystals Thermography.** Mater Eval 41(2):190+, 1983.

SOCIAL AND LIFE SCIENCES

Desarbo W. S., **Gennclus—New Models for General Nonhierarchical Clustering Analysis.** Psychometri 47(4):449–475, 1982.

Fishburn P. C., **Nontransitive Measurable Utility.** J Math Psyc 26(1):31–67, 1982.

Foley M. A., Johnson M. K., Raye C. L., **Age-Related-Changes in Confusion Between Memories for Thoughts and Memories for Speech.** Child Dev 54(1):51–60, 1983.

Hanley M. J., Wilkening G. M., **Nonionizing Radiations—Current Issues and Controversies—A Minisymposium (Editorial).** J Occup Med 25(2):95, 1983.

Petersen R. C., **Bioeffects of Microwaves—A Review of Current Knowledge.** J Occup Med 25(2):103–110, 1983.

Rosenbaum D. A., et al., **Hierarchical Control of Rapid Movement Sequences.** J Exp Psy P 9(1):86–102, 1983.

Stack L. C., Lannon P. B., Miley A. D., **Accuracy of Clinicians Expectancies for Psychiatric Rehospitalization.** Am J Comm P 11(1):99–113, 1983.

Starr S. J., **A Study of Video Display Terminal Workers.** J Occup Med 25(2):95–98, 1983.

Weiss M. M., **The Video Display Terminals—Is There a Radiation Hazard.** J Occup Med 25(2):98–100, 1983.

SPEECH/ACOUSTICS

Hall J. L., **A Procedure for Detecting Variability of Psychophysical Thresholds.** J Acoust So 73(2):663–667, 1983.

Pelech I., Zipfel G. G., Holford R. L., **A Wake-Scattering Experiment in Thermally Stratified Water.** J Acoust So 73(2):528–538, 1983.

CONTENTS, SEPTEMBER 1983

Part 1

Stabilized Biasing of Semiconductor Lasers
R. G. Swartz and B. A. Wooley

Empirical Evaluation of Profile Variations in an MCVD Optical Waveguide Fiber Using Modal Structure Analysis
A. Carnevale and U. C. Paek

A Comparison of Line Difference Predictions for Time-Frequency Multiplexing of Television Signals
R. L. Schmidt

On the Recognition of Isolated Digits From a Large Telephone Customer Population
J. G. Wilpon and L. R. Rabiner

Comparing Batch Delays and Customer Delays
W. Whitt

Batch Delays Versus Customer Delays
S. Halfin

Combined Source and Channel Coding for Variable-Bit-Rate Speech Transmission
D. J. Goodman and C.-E. Sundberg

Alternative Cell Configurations for Digital Mobile Radio Systems
C.-E. Sundberg

On Continuous Phase Modulation in Cellular Digital Mobile Radio Systems
C.-E. Sundberg

A Compatible High-Definition Television System
T. S. Rzeszewski

Part 3

TOTAL NETWORK DATA SYSTEM

Introduction
L. Schenker

Environment and Objectives
A. L. Barrese, D. E. Parker, T. E. Robbins, and L. M. Steele

System Plan
M. S. Hall, Jr., J. A. Kohut, G. W. Riesz, and J. W. Steifle

Theoretical and Engineering Foundations
W. S. Hayward and J. P. Moreland

Data Acquisition and Near-Real-Time Surveillance
C. J. Byrne, D. J. Gagne, J. A. Grandle, Jr., and G. H. Wedemeyer

Network Management
G. C. Ebner and D. G. Haenschke

National Network Management
W. S. Bartz and R. W. Patterson

Equipment Systems
N. D. Fulton, J. J. Galiardi, E. J. Pasternak, S. A. Schulman, and H. E. Voigt

Trunking Systems
P. V. Bezdek and J. P. Collins

Central Office Equipment Reports for Stored Program Control Systems
R. F. Grantges, V. L. Fahrmann, T. A. Gibson, and L. M. Brown

Small Office Network Data System
D. H. Barnes and J. J. O'Connor

Performance Measurement/Trouble Location
D. R. Anderson and M. J. Evans

Operating Company Perspective
J. Pfeiffer, Jr.

THE BELL SYSTEM TECHNICAL JOURNAL is abstracted or indexed by *Abstract Journal in Earthquake Engineering, Applied Mechanics Review, Applied Science & Technology Index, Chemical Abstracts, Computer Abstracts, Current Contents/ Engineering, Technology & Applied Sciences, Current Index to Statistics, Current Papers in Electrical & Electronic Engineering, Current Papers on Computers & Control, Electronics & Communications Abstracts Journal, The Engineering Index, International Aerospace Abstracts, Journal of Current Laser Abstracts, Language and Language Behavior Abstracts, Mathematical Reviews, Science Abstracts (Series A, Physics Abstracts; Series B, Electrical and Electronic Abstracts; and Series C, Computer & Control Abstracts), Science Citation Index, Sociological Abstracts, Social Welfare, Social Planning and Social Development*, and *Solid State Abstracts Journal*. Reproductions of the Journal by years are available in microform from University Microfilms, 300 N. Zeeb Road, Ann Arbor, Michigan 48106.


Bell System